

# Presolve of Linear Bilevel Programs

## Masterarbeit

zur Erlangung des Grades

**Master of Science (M.Sc.)**

**im Studiengang Mathematik**

am Department Mathematik der  
Friedrich-Alexander-Universität Erlangen-Nürnberg

vorgelegt am 02.03.2020

von **Julian Manns**

Betreuer: Prof. Dr. Frauke Liers,  
Dr. Dieter Weninger, Thomas Kleinert

---

# Contents

<b>1</b>	<b>Introduction to Linear Bilevel Optimization</b>	<b>3</b>
<b>2</b>	<b>Mathematical Modeling of Linear Bilevel Programs</b>	<b>4</b>
2.1	Components of a Linear Bilevel Program . . . . .	6
2.2	Approaches for Solving Linear Bilevel Programs . . . . .	9
2.2.1	KKT Reformulation . . . . .	9
2.2.2	<i>K</i> th-best Algorithm . . . . .	11
2.2.3	A Penalty-Alternating-Direction-Method-Based Primal Heuristic	12
2.2.4	Implementation . . . . .	13
<b>3</b>	<b>Dependence of Linear Bilevel Programs on Irrelevant Constraints</b>	<b>14</b>
3.1	Independence of Linear Programs of Irrelevant Constraints . . . . .	14
3.2	Implications of Independence of Irrelevant Constraints for Linear Bilevel Programs . . . . .	14
3.3	Independence of Local Optimality of Irrelevant Constraints for Linear Bilevel Programs . . . . .	18
<b>4</b>	<b>Theoretical Foundation for Presolve of Linear Bilevel Programs</b>	<b>23</b>
4.1	Independence of Inactive Constraints for Linear Programs . . . . .	24
4.2	Consequences of Removing Inactive Constraints and Adding Irrelevant Constraints . . . . .	27
4.3	Independence of Inactive Constraints for Linear Bilevel Programs . . . . .	30
4.4	Moving Inactive Constraints from the Follower to the Leader . . . . .	31
<b>5</b>	<b>Finding Inactive Constraints with the <i>K</i>th-best Algorithm</b>	<b>36</b>
<b>6</b>	<b>Presolve Techniques for Linear Bilevel Programs</b>	<b>40</b>
6.1	Bound Strengthening . . . . .	40
6.1.1	Bound Strengthening on Linear Programs . . . . .	40
6.1.2	Bound Strengthening on Linear Bilevel Programs . . . . .	42
6.2	Substitution . . . . .	45
6.3	Duality Fixing . . . . .	46
6.4	Parallel and Anti-Parallel Rows . . . . .	49
6.4.1	Presolve of Parallel Rows . . . . .	51

---

6.4.2	Presolve of Anti-Parallel Rows . . . . .	52
6.5	Rows that are Anti-Parallel to the Objective Coefficients . . . . .	52
6.6	Parallel Columns . . . . .	56
6.6.1	Both Columns represent Leader Variables . . . . .	58
6.6.2	Both Columns represent Follower Variables . . . . .	59
6.6.3	Columns with just Parallel Follower Coefficients . . . . .	60
<b>7</b>	<b>Conclusion and outlook</b>	<b>63</b>
	<b>References</b>	<b>66</b>

---

## Abstract

Bilevelprobleme sind komplexe, zweistufige Optimierungsprobleme, die verwendet werden können, um hierarchische Entscheidungsprozesse zu modellieren, wie sie z.B. in Energiemärkten, in der Planung militärischer Verteidigung von Infrastruktur oder in Pricing-Modellen vorkommen. Selbst die einfachste Art von Bilevelproblemen, bei denen nur lineare Zielfunktionen und Nebenbedingungen vorkommen, sind bereits nicht-konvexe Optimierungsprobleme. Äquivalente einstufige Formulierungen ersetzen das Optimierungsproblem der unteren Stufe durch seine nichtkonvexen Optimalitätsbedingungen. Daher sind lineare Bilevelprogramme von Natur aus schwierig zu lösen.

Das Vereinfachen von gemischt-ganzzahligen linearen Programmen, genannt Presolve, konnte das Lösen dieser Probleme signifikant beschleunigen. Jedoch gibt es nur sehr wenig Literatur, die sich mit dem Presolve von Bilevelproblemen befasst. In dieser Arbeit bereiten wir genannte Literatur im Kontext linearer Bilevelprobleme auf, leiten neue theoretische Grundlagen für das Presolven von linearen Bilevelprogrammen her. Wir verwenden diese Resultate, um zu analysieren, inwiefern gängige Presolve-Techniken für lineare und gemischt-ganzzahlige Programme auf linearen Bilevelprogramme angewandt werden können.

---

## Abstract

Bilevel programs are complex optimization problems that can be used to model hierarchical decision processes, which occur e.g. in energy markets, critical infrastructure defense or pricing models. Even the most simple bilevel programs, where only linear objective functions and constraints appear, are non-convex optimization problems and equivalent single level formulations replace the lower level problem by its non-convex optimality constraints. This makes linear bilevel programs inherently difficult to solve.

The simplification of mixed-integer linear programs before solving them, called presolve, significantly accelerated the solving of these problems. However, there is only very few literature on the topic of presolve of bilevel programs. In this thesis we review said literature on presolve of bilevel programs in the context of linear bilevel programming, derive new theoretical foundations for presolve of linear bilevel programs and then apply these results to analyze how common presolve techniques for linear and mixed integer programs can be used to presolve linear bilevel programs.

---

# 1 Introduction to Linear Bilevel Optimization

Bilevel programs are optimization problems where the ultimate outcome also depends on the reaction of another party rather than just your own decision. In this sense, the goal is to provoke the reaction that is most beneficial for yourself. These problems are also called Stackelberg games [30, 29], where the initial decision is made by the leader and the reaction made by the follower. The reaction of the follower is assumed to be deterministic based on the leader's decision, such that the leader can anticipate the reaction. Practical applications that can be modeled with bilevel programs occur for example in energy markets [2, 10, 17, 18, 19, 20, 22, 25], critical infrastructure defense [9, 13] and pricing models [26].

In bilevel programs the goals of the leader and the follower are expressed through single-level optimization problems. In the simplest case, i.e. where both the leader and the follower problem are linear programs, the bilevel program is already NP hard, as shown in [11, 21]. Approaches for solving bilevel programs include, but are not limited to, reformulations into single level mixed integer programs [15], reversed search tree algorithms [5] and heuristic approaches for finding local optima [24].

In order to improve the computational performance of optimization problems of different kinds, numerous presolve techniques have been developed. The presolve techniques that have the most impact on linear programs but also work well on mixed integer programs reduce the size, i.e. the number of variables and constraints of the model, as well as improve numerical stability, for example by removing parallel or almost parallel rows and columns. On top of that, presolve techniques for mixed integer programs also aim to improve the formulation of the model in order to bring the polyhedron described by the constraints closer to the actual convex hull of the feasible points. In [3] we see how beneficial presolve can be for linear programs and in [1, 8, 6] we see the significant improvements on solvers for mixed integer programs due to presolve.

However, to the best of our knowledge, no presolve techniques for bilevel programs have been developed. In [27, 12] some fundamental properties of optimization problems, that enable certain single level presolve techniques and other single level solver improvements such as so called Cutting Planes Methods, have been investigated with the result that these properties are generally not present in bilevel programs.

The outline of this thesis is as follows. In Section 2 we present the basic formulation of linear bilevel programs and their components, as well as three approaches for finding solutions to the programs. In Section 3 we review the existing, yet sparse, literature on the presolve of linear bilevel programs. In Section 4 we present essential theoretical results on the modification of linear bilevel programs, which we will use in Section 6 to prove the correctness of various heuristic presolve approaches. Section 5 contains a heuristic method that uses information gathered by the  $K$ th-best algorithm to presolve the linear bilevel program.

---

## 2 Mathematical Modeling of Linear Bilevel Programs

In order to model the hierarchical structure of bilevel programs the variables will be split into two subsets, namely the vectors  $x$  and  $y$ , where the leader operates  $x$  and the follower operates  $y$ . Both the leader's and the follower's goals are formulated as optimization problems, the follower problem being a parametrized problem with the leader's decision  $x$  as a parameter.

**Example 2.1.** Consider the parameterized linear program

$$\min_{y \in \mathbb{R}} y \quad (2.1a)$$

$$s. t. \quad y \geq 4 - 2x, \quad (2.1b)$$

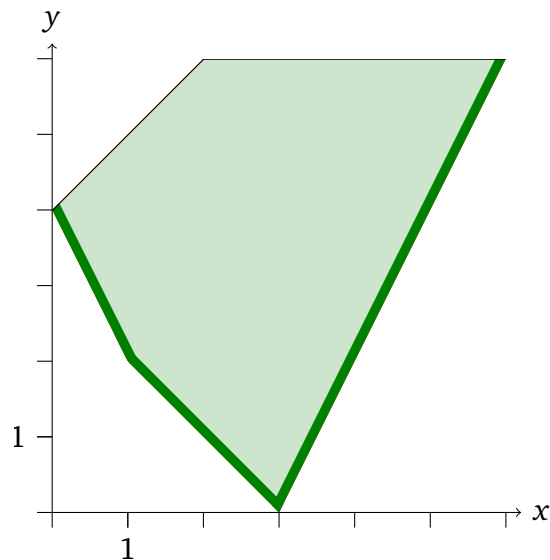
$$y \geq 3 - x, \quad (2.1c)$$

$$y \geq -6 + 2x, \quad (2.1d)$$

$$y \leq 4 + x, \quad (2.1e)$$

$$y \leq 6, \quad (2.1f)$$

where  $x \in \mathbb{R}$  is given. We can visualize this program the same way we would visualize a regular two dimensional linear program, but instead of a single optimal solution we get an optimal solution for each possible value of  $x$ :



The thick green outline shows all points optimal for (2.1). We now create a linear bilevel program from this by maximizing  $4y - x$  among all  $(x, y) \in \mathbb{R}^2$  where  $y$  is optimal for

the linear program (2.1). This leads to the model

$$\max_{x,y} 4y - x \quad (2.2a)$$

$$\text{s. t. } y \in \operatorname{argmin}_{\bar{y}} \bar{y} \quad (2.2b)$$

$$\text{s. t. } \bar{y} \geq 4 - 2x, \quad (2.2c)$$

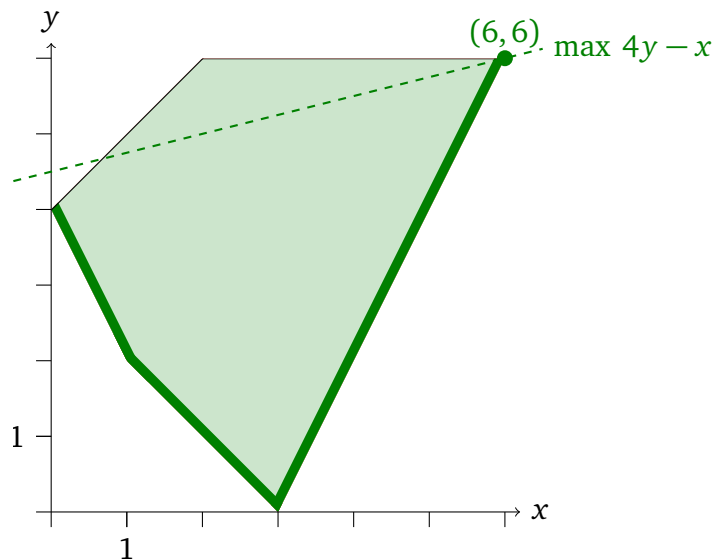
$$\bar{y} \geq 3 - x, \quad (2.2d)$$

$$\bar{y} \geq -6 + 2x, \quad (2.2e)$$

$$\bar{y} \leq 4 + x, \quad (2.2f)$$

$$\bar{y} \leq 6. \quad (2.2g)$$

We can now draw this new objective function  $4y - x$  into the figure to get



and we already see that this linear bilevel program is not convex, since the green outline represents the points feasible to (2.2). For example, the point  $(0, 4)$  is a local but not a global optimal solution. We now introduce two leader constraints:

$$\max_{x,y} 4y - x \quad (2.3a)$$

$$\text{s. t. } y \geq 1, \quad (2.3b)$$

$$y \leq 5, \quad (2.3c)$$

$$y \in \operatorname{argmin}_{\bar{y}} \bar{y} \quad (2.3d)$$

$$\text{s. t. } \bar{y} \geq 4 - 2x, \quad (2.3e)$$

$$\bar{y} \geq 3 - x, \quad (2.3f)$$

$$\bar{y} \geq -6 + 2x, \quad (2.3g)$$

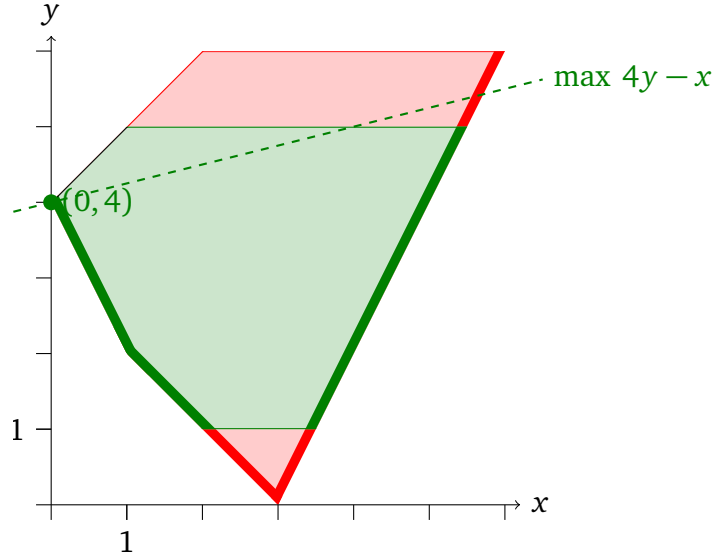
$$\bar{y} \leq 4 + x, \quad (2.3h)$$

$$\bar{y} \leq 6. \quad (2.3i)$$



---

Visualized, some sections of the former green outline are now not feasible anymore:



The thick outline (red and green) now describes all points optimal for the follower while the green part are the ones also feasible for the leader. We can see that this thick green outline is not only not convex, but not even connected.

## 2.1 Components of a Linear Bilevel Program

A general linear bilevel program can be written in the form

$$\max_{x,y} c^\top x + d^\top y \quad (2.4a)$$

$$\text{s. t. } Ax + By \leq a, \quad (2.4b)$$

$$y \in \arg \min_{\bar{y}} e^\top \bar{y} \quad (2.4c)$$

$$\text{s. t. } D\bar{y} \geq b - Cx. \quad (2.4d)$$

This definition of a bilevel program goes by the *optimistic assumption* because we assume that whenever the follower problem has multiple optimal solutions, the one that is best for the leader will be chosen. In the course of this thesis we will only cover optimistic linear bilevel programs.

The model has these basic components:

- $x \in \mathbb{R}^n$  are the *leader variables*.
- $c^\top x + d^\top y$  is the *leader objective* with  $c \in \mathbb{R}^n$  and  $d \in \mathbb{R}^m$ .
- $Ax + By \leq a$  are the *leader constraints* with  $A \in \mathbb{R}^{k \times n}$ ,  $B \in \mathbb{R}^{k \times m}$  and  $a \in \mathbb{R}^k$ .

- 
- $y$  and  $\bar{y}$  are the *follower variables*.
  - $e^\top y$  is the *follower objective* with  $e \in \mathbb{R}^m$ .
  - $Dy \geq b - Cx$  are the *follower constraints* with  $C \in \mathbb{R}^{l \times n}$ ,  $D \in \mathbb{R}^{l \times m}$  and  $b \in \mathbb{R}^l$ .

We say that the program has  $n$  leader variables (leader columns),  $m$  follower variables (follower columns),  $k$  leader constraints (leader rows) and  $l$  follower constraints (follower rows).

We will now declare several other components of (2.4) that will be important throughout the course of this thesis:

- The parametric linear program

$$\min_y e^\top y \quad (2.5a)$$

$$\text{s. t. } Dy \geq b - Cx, \quad (2.5b)$$

where  $x$  is given, is called the *follower problem*. The set

$$\Omega_F = \{(x, y) \mid Cx + Dy \geq b\}$$

is called the *follower's constraint region* or *follower polyhedron*. This is not exactly the feasible set of (2.5), since the follower problem is parametric in  $x$  and only operates on  $y$ . Analogously, we define

$$\Omega_L = \{(x, y) \mid Ax + By \leq a\}$$

as the *leader's constraint region* or *leader polyhedron*.

- The set of optimal solutions of the follower problem

$$\Psi(x) := \arg \min_y \{e^\top y \mid Dy \geq b - Cx\}$$

is functionally dependent on  $x$  and is called the *rational reaction set*. The linear bilevel program (2.4) can then equivalently be formulated as

$$\max_{x,y} c^\top x + d^\top y \quad (2.6a)$$

$$\text{s. t. } Ax + By \leq a, \quad (2.6b)$$

$$y \in \Psi(x). \quad (2.6c)$$

In the plots in Example (2.3) the rational reaction set for each choice of  $x$  is represented by the red and green outline.

- The set of feasible points of (2.4) is given by

$$IR := \{(x, y) \in \mathbb{R}^{n \times m} \mid Ax + By \leq a, y \in \Psi(x)\}$$

and is called the *inducible region* or the set of bilevel feasible points. In Example 2.1 we saw that the inducible region is generally not convex and indeed not even connected, which makes linear bilevel programs generally non-convex. In the plots in Example 2.1 the rational reaction set is represented by the green outline.

- Beside the follower problem, another important linear program is associated with (2.4):

$$\max_{x,y} c^\top x + d^\top y \quad (2.7a)$$

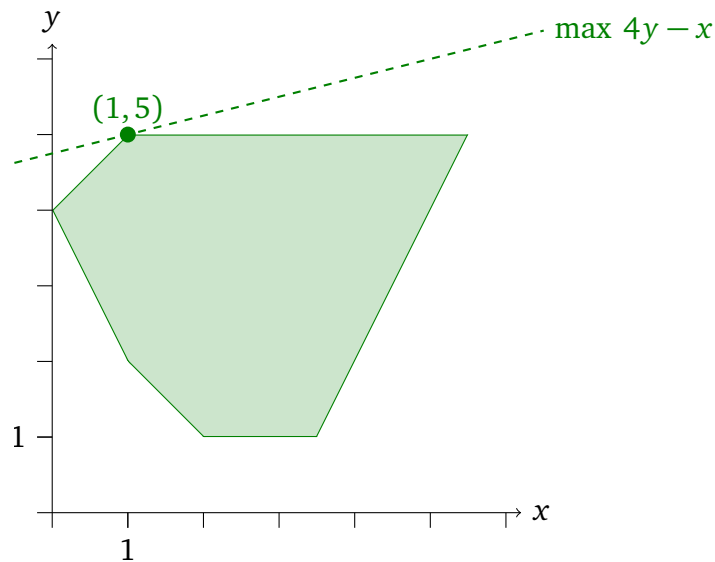
$$\text{s. t. } Ax + By \leq a, \quad (2.7b)$$

$$Cx + Dy \geq b \quad (2.7c)$$

Starting from the formulation (2.6), the constraint  $y \in \Psi(x)$  is replaced by  $Cx + Dy \geq b$ . Since  $Cx + Dy \geq b$  is part of the follower problem,  $y \in \Psi(x)$  implies  $Cx + Dy \geq b$ . Hence, (2.7) is rightfully called the *high point relaxation* and its feasible set

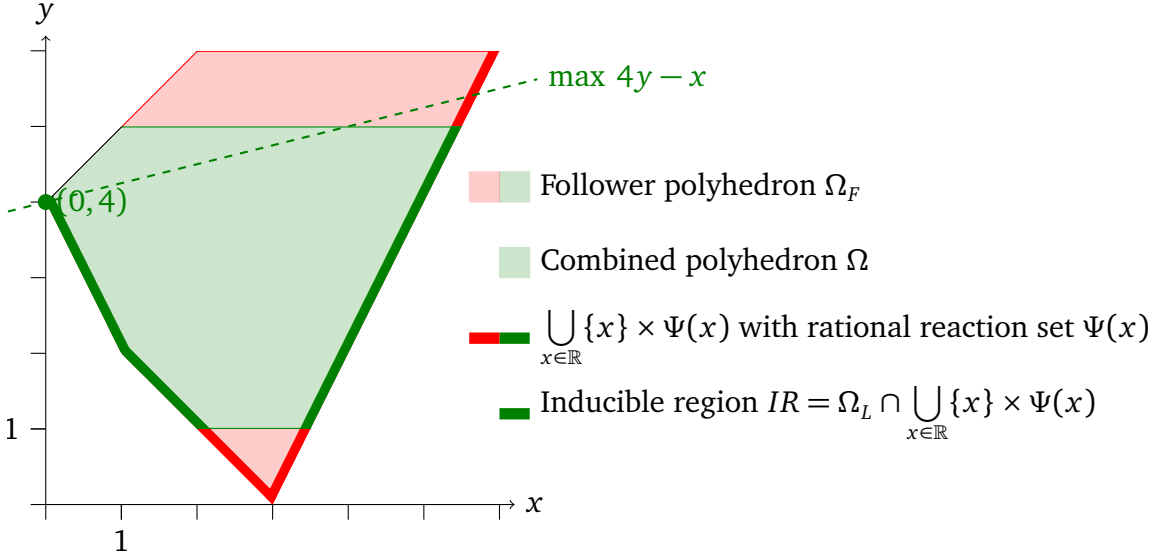
$$\Omega = \Omega_L \cap \Omega_F = \{(x, y) \mid Ax + By \leq a, Cx + Dy \geq b\}$$

is called the *combined constraint region* or *combined polyhedron*. The high point relaxation of (2.3) looks like this:



Here we already see that the high point relaxation can generally have a better solution than the original bilevel program.

We summarize these components in the following figure:



## 2.2 Approaches for Solving Linear Bilevel Programs

We will now present three different approaches for finding optimal or feasible solutions to linear bilevel programs.

### 2.2.1 KKT Reformulation

The Karush-Kuhn-Tucker-constraints (KKT) can be used to reformulate a given linear bilevel program into a linear mixed integer single-level program (MIP), which can be solved using common MIP-solvers. As far as we know, this approach for solving linear bilevel programs generally performs the best. In this section we will present the KKT reformulation in order to understand which components of the linear bilevel program are "more expensive" than others.

We reformulate the linear bilevel program given by (2.4). The dual of the follower problem (2.5) is

$$\max_{\lambda} \quad (b - Cx)^T \lambda, \quad (2.8a)$$

$$\text{s.t.} \quad D^T \lambda = e, \quad (2.8b)$$

$$\lambda \geq 0. \quad (2.8c)$$

By applying strong duality we can replace the follower problem (2.4c) in (2.4) by the primal and dual constraints of the follower problem together with the complementarity

---

ity constraint and get the following single level reformulation of (2.4):

$$\max_{x,y,\lambda} \quad c^T x + d^T y, \quad (2.9a)$$

$$\text{s.t.} \quad Ax + By \leq a, \quad (2.9b)$$

$$Cx + Dy \geq b, \quad (2.9c)$$

$$D^T \lambda = e, \quad (2.9d)$$

$$\lambda \geq 0, \quad (2.9e)$$

$$(C_j x + D_j y - b_j) \lambda_j = 0 \quad \forall j. \quad (2.9f)$$

While this is single level, it is not convex because the complementarity constraints (2.9f) are bilinear and non-convex. However, these complementarity constraints can be reformulated into linear integer constraints, as presented in [15]. The meaning of

$$(C_j x + D_j y - b_j) \lambda_j = 0$$

is that

$$C_j x + D_j y - b_j = 0 \quad \vee \quad \lambda_j = 0.$$

We can introduce a binary variable  $s_j$  that controls which of these two terms has to be 0. To do this, we need an arbitrarily large number  $M_j$  of which we know that neither  $C_j x + D_j y - b_j$  nor  $\lambda_j$  are going to be bigger than  $M_j$  when looking at an optimal solution. These large-enough  $M$ s are bluntly called big- $M$ s. Then we can reformulate (2.9f) into

$$\begin{aligned} 0 \leq C_j x + D_j y - b_j &\leq M_j s_j, \\ 0 \leq \lambda &\leq M_j (1 - s_j), \\ s_j &\in \{0, 1\}. \end{aligned}$$

This forces one of the terms to be 0, depending on the state of  $s_j$ . Note that the non-negativity constraints on the left are already provided by (2.9b) and (2.9e). Now we can summarize the reformulation:

$$\max_{x,y,\lambda} \quad c^T x + d^T y, \quad (2.10a)$$

$$\text{s.t.} \quad Ax + By \leq a, \quad (2.10b)$$

$$Cx + Dy \geq b, \quad (2.10c)$$

$$D^T \lambda = e, \quad (2.10d)$$

$$\lambda \geq 0, \quad (2.10e)$$

$$C_j x + D_j y - b_j \leq M_j s_j, \quad (2.10f)$$

$$\lambda_j \leq M_j (1 - s_j), \quad (2.10g)$$

$$s_j \in \{0, 1\}. \quad (2.10h)$$

This is a linear mixed integer program that is, in theory, solvable by common solvers. However, the selection of the big- $M$ s poses problems. First of all, they cannot just be as big as possible since too large big- $M$ s cause numerical problems. This narrows

---

down the feasible interval for the big- $M$ s. In [28] it is shown that the wrong selection of big- $M$ s can lead to highly suboptimal solutions. Furthermore, in [23] it is shown that finding appropriate big- $M$ s can be as hard as solving the non-reformulated model itself. The practical solvability of this reformulated model heavily relies on a good way of determining the big- $M$ s through the structure of the problem this model was based on.

Next up we will analyze the dimensions of (2.10) depending on the dimensions of the original linear bilevel program, which has  $n$  leader variables,  $m$  follower variables,  $k$  leader constraints and  $l$  follower constraints. (2.10) has  $n+m+l$  continuous variables,  $l$  binary variables and  $k+l+m+3l = m+k+4l$  linear constraints. To summarize:

- Each leader variable produces one continuous variable ( $x$ ).
- Each leader constraint produces one linear constraint (2.10b).
- Each follower variable produces one continuous variable ( $y$ ) and one linear constraint (2.10d).
- Each follower constraint produces one linear variable ( $\lambda$ ), one integer variable ( $s$ ) and four linear constraints (2.10c), (2.10e), (2.10f) and (2.10g).

Here we can see clearly that, with respect to model dimensions, leader variables as well as leader constraints produce no additional variables or constraints while follower variables and follower constraints do so. The follower constraints impact the reformulated model size more than any of the other three components.

With respect to non-zeroes in the coefficients, we see that  $e$  matters twice as much as  $c$  and  $d$ ,  $C$  matters twice as much as  $A$  and  $B$  and  $D$  matters three times as much as  $A$  and  $B$ .

This analysis is important in the context of presolve since it shows removing which components through presolve is most beneficial. It also justifies the following corollary, which we will frequently refer to during the course of this thesis.

**Remark 2.2.** *When solving a linear bilevel program using the KKT-reformulation, moving variables and constraints from the follower to the leader beforehand decreases the size of the KKT-reformulated model.*

### 2.2.2 $K$ th-best Algorithm

The  $K$ th-best algorithm, as proposed in [5], traverses the extreme points of the high point relaxation, starting with the best and then descending with respect to the objective value, until it finds an extreme point that is bilevel feasible. Consequently, this point must also be bilevel optimal. We will explain this algorithm in detail in Section 5.

---

### 2.2.3 A Penalty-Alternating-Direction-Method-Based Primal Heuristic

Penalty alternating direction methods (PADM) are algorithms that attempt to solve an optimization program by splitting the variables into two sets, for example into  $x$  and  $\lambda$ . Then, e.g.  $x$  is fixed to some initial value and the program is solved for  $\lambda$  resulting in a solution  $\lambda_0$ . Afterwards  $\lambda = \lambda_0$  is fixed and the program is solved for  $x$  resulting in  $x_0$  which is then used to solve for  $\lambda$  again resulting in  $\lambda_1$  and so on.

In [24] a heuristic which employs a penalty alternating direction method to find bilevel feasible points in a bilevel program is developed and analyzed. It uses a single-level reformulation similar to (2.9), but instead of the complementary constraints (2.9f) it enforces optimality through the strong duality constraint  $e^\top y = (b - Cx)^\top \lambda$ . Since  $e^\top y \geq (b - Cx)^\top \lambda$  is already given by weak duality for linear programs, we only need to add  $e^\top y \leq (b - Cx)^\top \lambda$  to enforce strong duality. This leads to the reformulation

$$\max_{x,y,\lambda} \quad c^\top x + d^\top y, \quad (2.11a)$$

$$\text{s.t.} \quad Ax + By \leq a, \quad (2.11b)$$

$$Cx + Dy \geq b, \quad (2.11c)$$

$$D^\top \lambda = e, \quad (2.11d)$$

$$\lambda \geq 0, \quad (2.11e)$$

$$e^\top y \leq (b - Cx)^\top \lambda. \quad (2.11f)$$

The strong duality constraint (2.11f) is then relaxed by introducing it into the objective function as a penalty term (hence the *penalty* in PADM):

$$\max_{x,y,\lambda} \quad c^\top x + d^\top y + \rho(e^\top y - (b - Cx)^\top \lambda), \quad (2.12a)$$

$$\text{s.t.} \quad Ax + By \leq a, \quad (2.12b)$$

$$Cx + Dy \geq b, \quad (2.12c)$$

$$D^\top \lambda = e, \quad (2.12d)$$

$$\lambda \geq 0. \quad (2.12e)$$

The penalty term is weighted with  $\rho > 0$ , which will be increased during the process of solving (more on that later on). Now the variables are split into the subsets  $(x, y)$  and  $\lambda$  and an alternating direction method is applied by alternatively solving

$$\lambda^i \in \arg \min_{\lambda} \quad (b - Cx^i)^\top \lambda, \quad (2.13a)$$

$$\text{s.t.} \quad D^\top \lambda = e, \quad (2.13b)$$

$$\lambda \geq 0, \quad (2.13c)$$

and

$$(x^{i+1}, y^{i+1}) \in \arg \max_{x,y} \quad c^\top x + d^\top y + \rho(e^\top y - (b - Cx)^\top \lambda^i), \quad (2.14a)$$

$$\text{s.t.} \quad Ax + By \leq a, \quad (2.14b)$$

$$Cx + Dy \geq b \quad (2.14c)$$

---

for some initial values  $(x^0, y^0)$ . This leads to a series  $(x^i, y^i)$ . While [24] provides no hard proof that  $(x_i, y_i)$  converges at all, it is shown, however, that if it converges then it converges to a bilevel feasible point. Precisely how  $\rho$  needs to be increased in order to achieve this conversion is described in [24]. The evaluation of this method showed that it is very likely to converge and generally provides a good solution, in numerous cases even an optimal solution.

Similarly to Remark 2.2, we can argue that the size of model (2.13) depends on the size of the follower problem (2.5). It is therefore more beneficial to reduce the size of the follower than the size of the leader.

#### **2.2.4 Implementation**

Both the KKT reformulation and the PADM heuristic have been implemented by Martin Schmidt and Thomas Kleinert in a C++ framework as solvers for bilevel programs. These solvers were used for testing several techniques. However, the thesis does not cover a computational evaluation of the presented techniques.



---

### 3 Dependence of Linear Bilevel Programs on Irrelevant Constraints

The only existing literature that is related to the presolve of linear bilevel programs are the different publications on the topic of *independence of irrelevant constraints*, or short, *IIC*. In this chapter we will first examine what independence of irrelevant constraints is, what it has to do with presolve and why it is taken for granted when working with single level programs, as stated in [27]. Afterwards, we present the most significant publications on independence of irrelevant constraints and put them in the context of linear bilevel programs.

#### 3.1 Independence of Linear Programs of Irrelevant Constraints

Consider a general linear program of the form

$$\max_x \quad c^T x, \tag{3.1a}$$

$$\text{s.t.} \quad Ax \leq b. \tag{3.1b}$$

Furthermore, let  $a_0^T x \leq b_0$  be an arbitrary new constraint and

$$\max_x \quad c^T x, \tag{3.2a}$$

$$\text{s.t.} \quad Ax \leq b, \tag{3.2b}$$

$$a_0^T x \leq b_0 \tag{3.2c}$$

a modified version of (3.1) that includes the new constraint. It is easy to see that every optimal solution  $x^*$  of (3.1) that satisfies

$$a_0^T x \leq b_0$$

is also an optimal solution to (3.2). Also, no new optimal solutions can be found in (3.2). Hence, adding a constraint to (3.1) that is satisfied by all optimal solutions of (3.1), namely an *irrelevant constraint*, will result in a modified (3.2) with the same set of optimal solutions. In that sense, linear programs are independent of irrelevant constraints.

#### 3.2 Implications of Independence of Irrelevant Constraints for Linear Bilevel Programs

While linear programs are IIC, this is generally not the case for bilevel programs. While it is easy to see that irrelevant constraints can be added to the leader of a bilevel program since that will only remove non-optimal solutions from the inducible region, they can however change the outcome of the program when they are added to the follower problem. The following example illustrates this.

**Example 3.1.** Consider the linear bilevel program

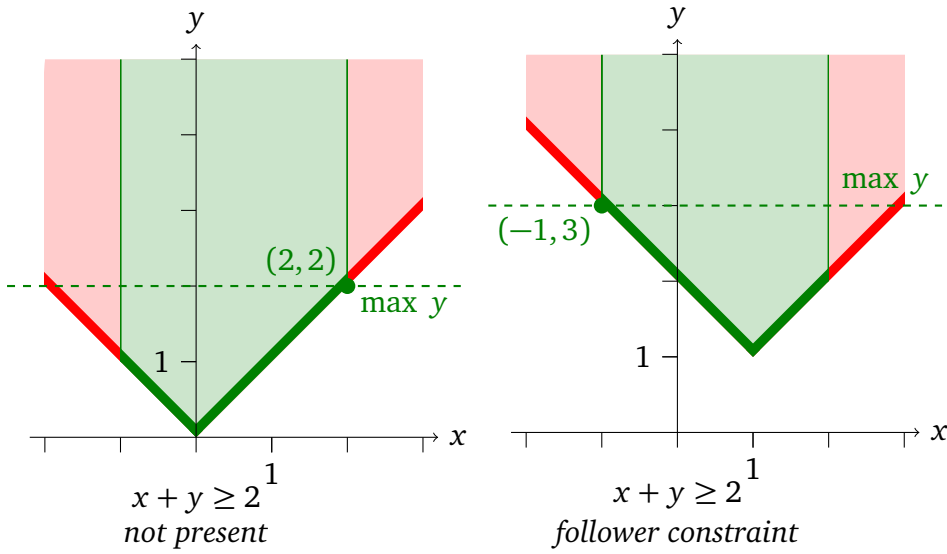
$$\max_{x,y} y \quad (3.3a)$$

$$\text{s. t. } -1 \leq x \leq 2, \quad (3.3b)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (3.3c)$$

$$\text{s. t. } \bar{y} \geq x, \quad (3.3d)$$

$$\bar{y} \geq -x. \quad (3.3e)$$



which has the optimal solution  $(2, 2)$ . The arbitrary constraint  $x + y \geq 2$  is satisfied by this optimal solution. However, adding this constraint to the follower limits the follower in such a way that he now responds with  $y = 3$  for  $x = -1$ , which is a better solution for the leader.

In paper [27] it was established that independence of irrelevant constraints is a very restrictive property that makes the problem trivial to a certain degree. In this subsection we will put the contents of said paper in the context of linear bilevel programming.

We consider the linear bilevel program  $P$  given by (2.4). Let  $\tilde{P}(u, v, w)$  be the modified version of  $P$  given by

$$\max_{x,y} c^\top x + d^\top y \quad (3.4a)$$

$$\text{s. t. } Ax + By \leq a, \quad (3.4b)$$

$$y \in \arg \min_{\bar{y}} e^\top \bar{y} \quad (3.4c)$$

$$\text{s. t. } D\bar{y} \geq b - Cx, \quad (3.4d)$$

$$v^\top \bar{y} \geq w - u^\top x. \quad (3.4e)$$

---

for any  $u \in \mathbb{R}^n$ ,  $v \in \mathbb{R}^m$ ,  $w \in \mathbb{R}$ . We call  $u^\top x + v^\top y \geq w$  an arbitrary constraint. Furthermore, let  $Opt(P)$  and  $Opt(\tilde{P}(u, v, w))$  be the sets of optimal solutions of  $P$  and  $\tilde{P}(u, v, w)$  respectively.

Now we define independence of irrelevant constraints:

**Definition 3.2.**  $P$  is independent of irrelevant constraints (IIC), if

$$(x^*, y^*) \in O(\tilde{P}(u, v, w)) \quad \forall (x^*, y^*) \in O(P) : u^\top x^* + v^\top y^* \geq w, \quad \forall u, v, w.$$

This means that a linear bilevel program is IIC if, after modifying the program by adding any arbitrary constraint to the follower, every optimal solution of the original program that satisfies the arbitrary constraint is also an optimal solution of the modified program.

At this point it is important to note that in [27] a bilevel program (which could be a linear bilevel program) is IIC, if any irrelevant constraint, and not just irrelevant *linear* constraints, can be added without changing the optimal solution. However, later on we will see that these two variants for the definition of IIC are equivalent.

To investigate necessary and sufficient conditions for a bilevel problem to be IIC, [27] introduces the unconstrained follower problem, which is the problem of minimizing the follower objective (2.4c) without any of the constraints (2.4d):

$$\min_y e^\top y. \tag{3.5}$$

Obviously, this is a rather redundant problem in the linear case: If  $e^\top \neq 0$ , then (3.5) is unbounded (and therefore unsolvable). If  $e^\top = 0$ , then any solution  $y \in \mathbb{R}^m$  is optimal. [27] then defines bilevel problems as *degenerate* if a solution  $(x, y)$  to its high point relaxation exists, for which the gradient of the follower objective equals zero. However, a gradient of zero in the linear case means  $e^\top = 0$ . This leads to the following definition:

**Definition 3.3.** The linear bilevel program (2.4) is *degenerate*, if the high point relaxation is infeasible (which makes (2.4) infeasible in the first place) or if

$$e^\top = 0.$$

If we apply  $e^\top = 0$  to (2.4) then (2.4c) becomes redundant and we can rewrite the problem to

$$\max_{x, y} c^\top x + d^\top y \tag{3.6a}$$

$$\text{s. t. } Ax + By \leq a, \tag{3.6b}$$

$$Cx + Dy \geq b. \tag{3.6c}$$

which is exactly the high point relaxation (2.7). It follows that a degenerate linear bilevel program can be replaced by its high point relaxation, since either both are unsolvable or both have the same solution. Since the high point relaxation is a single

---

level problem, it is also IIC and therefore all degenerate linear bilevel programs are IIC.

For the next theorem, [27] only considers bilevel problems where the KKT conditions of the follower problem are necessary and sufficient for its optimality. In the linear case, the follower problem is a linear program and all linear programs satisfy this KKT requirement.

This leaves us with the following linear version of Theorem 1 of [27]:

**Theorem 3.4.** *If (2.4) is not degenerate then it is IIC if and only if there is a solution to the high point relaxation (2.7) that is feasible to (2.4).*

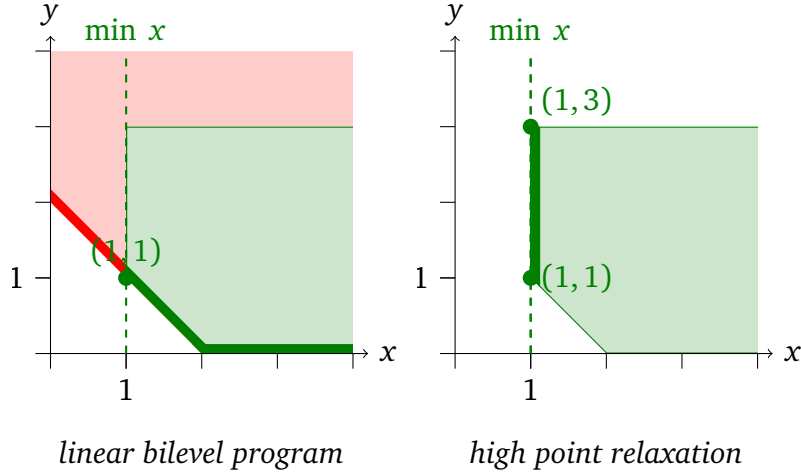
Technically, the version of this theorem in [27] requires the supposedly stronger definition for IIC that also includes irrelevant non-linear constraints. However, the proof of this theorem only makes use of irrelevant linear constraints to show that an IIC program has a solution to the high point relaxation that is feasible to the bilevel program. Thus, defining the IIC property only with linear constraints is equivalent to the definition with general constraints.

As the high point relaxation (2.7) is a relaxation of (2.4), every solution to (2.7) that is feasible to (2.4) also is an optimal solution to (2.4). This means a linear bilevel program that satisfies Theorem 3.4 can be solved by finding an appropriate solution of the high point relaxation.

The following example shows that a linear bilevel program that can be solved by one of the solutions of its high point relaxation, cannot necessarily be solved by every solution of its high point relaxation.

**Example 3.5.** *The linear bilevel program*

$$\begin{aligned}
 \min_{x,y} \quad & x \\
 \text{s. t.} \quad & x \geq 1, \\
 & y \leq 3, \\
 & y \in \arg \min_{\bar{y}} \bar{y} \\
 & \text{s. t.} \quad \bar{y} \geq 2 - x, \\
 & \bar{y} \geq 0
 \end{aligned}$$



has the optimal solution  $(1, 1)$  while its high point relaxation

$$\begin{aligned} \min_{x,y} \quad & x \\ \text{s. t.} \quad & x \geq 1, \\ & y \leq 3, \\ & y \geq 2 - x, \\ & y \geq 0 \end{aligned}$$

has optimal solutions

$$\{(1, y) \mid 1 \leq y \leq 3\}.$$

### 3.3 Independence of Local Optimality of Irrelevant Constraints for Linear Bilevel Programs

In this chapter [12] it was derived that the local optimality of solutions of bilevel programs is independent of irrelevant constraints if certain requirements are met. We will put this paper in the context of linear bilevel programming and we will show that these requirements are met by linear bilevel programs.

We consider the linear bilevel program (2.4) and the modified version (3.4) that has an additional follower constraint  $u^\top x + v^\top y \leq w$  with  $u \in \mathbb{R}^n$ ,  $v \in \mathbb{R}^m$ ,  $w \in \mathbb{R}$ . Furthermore, let  $(x^*, y^*)$  be an optimal solution to (2.4) and let  $u^\top x^* + v^\top y^* > w$ . Theorem 2.2 of [12] states that  $(x^*, y^*)$  is a local optimal solution to (3.4) if  $\Psi(x)$  is *inner semicontinuous*, which is defined as

**Definition 3.6.** Let  $x_0 \in \mathbb{R}^n$  and  $y_0 \in \Psi(x_0)$ . The mapping  $x \mapsto \Psi(x)$  is *inner semicontinuous* at  $(x_0, y_0)$  if for every sequence  $(x^i)$  with  $\Psi(x^i) \neq \emptyset$ , which converges to  $x_0$ , there is a sequence  $(y^i)$  with  $y^i \in \Psi(x^i)$  that converges to  $y_0$ .

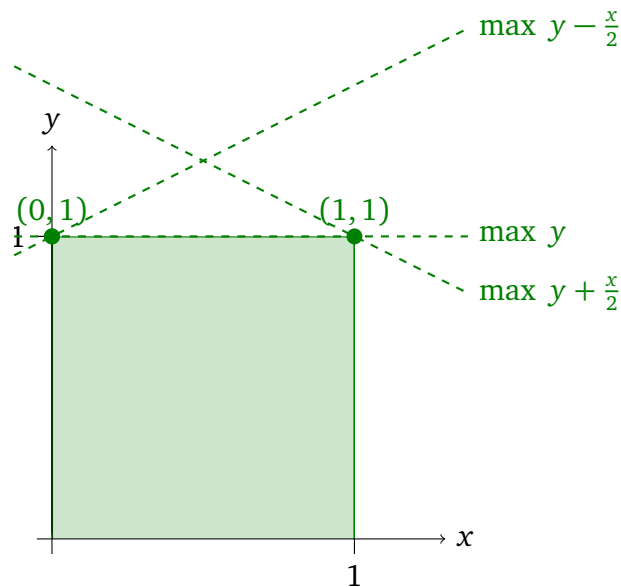
We have no hard proof that the rational reaction set mapping  $x \mapsto \Psi(x)$  is inner semicontinuous except for that in [31] it is shown that  $x \mapsto \Psi(x)$  is inner semicontinuous

(even continuous) if  $x$  is 1-dimensional, which suggests that  $\Psi(x)$  is generally continuous. Note that it is important that the follower problem is only parameterized in the right hand sides. We will now present examples that show that the mapping of the parameter to the optimal solution set is generally not continuous for a parameterized linear program, if the left hand sides or the objective function are parameterized.

**Example 3.7.** Consider the linear program

$$\begin{aligned} \max_{x,y} \quad & \theta x + y, \\ \text{s.t.} \quad & x \leq 1, \\ & y \leq 1, \\ & x \geq 0. \end{aligned}$$

with a parameterized objective function.

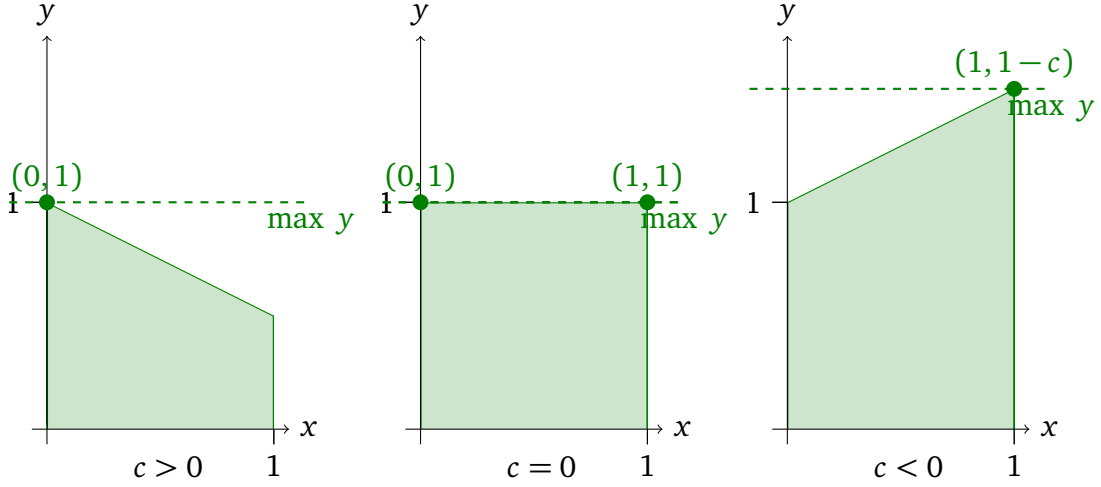


For  $\theta < 0$  the solution set is  $\{(1, 1)\}$ , for  $\theta = 0$  it is  $[0, 1] \times \{1\}$  and for  $\theta > 0$  it is  $\{(0, 1)\}$ . That means for any  $x \in (0, 1)$  and any sequence  $\theta^i \rightarrow 0$  with  $\theta_i \neq 0$  the optimal solution sequence  $(x^i, y^i)$  does not converge to  $(x, 0)$  because  $(x^i, y^i) = (0, 1)$  or  $(x^i, y^i) = (1, 1)$  for all  $i$ .

**Example 3.8.** Consider the linear program

$$\begin{aligned} \max_{x,y} \quad & y, \\ \text{s.t.} \quad & x \leq 1, \\ & \theta x + y \leq 1, \\ & x \geq 0. \end{aligned}$$

with a parameterized left hand side.



For  $c < 0$  the solution set is  $\{(1, 1 - c)\}$ , for  $c = 0$  it is  $[0, 1] \times \{1\}$  and for  $c > 0$  it is  $\{(0, 1)\}$ . That means for any  $x \in (0, 1)$  and any sequence  $\theta^i \rightarrow 0$  with  $\theta_i \neq 0$  the optimal solution sequence  $(x^i, y^i)$  does not converge to  $(x, 0)$  because  $(x^i, y^i) = (0, 1)$  or  $(x^i, y^i) = (1, 1 - c)$  for all  $i$ .

While we do not have the continuity of  $\Psi(x)$  to work with, we do however know that the inducible region is piecewise linear as shown in [4], which we can use to provide an alternative proof for

**Theorem 3.9.** *Let  $(x^*, y^*)$  be an optimal solution to (2.4) that satisfies  $u^\top x^* + v^\top y^* > w$ . Then  $(x^*, y^*)$  is a local optimal solution of (3.4).*

*Proof.* According to Theorem 4.12,  $(x^*, y^*)$  is feasible to (3.4). We now assume that  $(x^*, y^*)$  was not locally optimal in (3.4). Since the inducible region  $\widetilde{IR}$  of (3.4) is piecewise linear, as shown in [4], there must be a  $(x_0, y_0) \in \widetilde{IR}$  with

$$c^\top x_0 + d^\top y_0 < c^\top x^* + d^\top y^*. \quad (3.7)$$

and

$$\theta(x_0, y_0) + (1 - \theta)(x^*, y^*) \in \widetilde{IR} \quad \forall 0 \leq \theta \leq 1 \quad (3.8)$$

Furthermore,

$$u^\top x + v^\top y > w \quad (3.9)$$

with  $(x, y) = (x_0, y_0) + (1 - \theta)(x^*, y^*)$  is fulfilled for  $\theta = 0$  it also must be fulfilled for a sufficiently small  $\theta > 0$  (because of continuity). Since  $(x, y) \in \widetilde{IR}$ ,  $y$  is optimal for the follower problem of (3.4) given the leader choice  $x$  and  $u^\top x + v^\top y > w$  is inactive. Therefore,  $y$  is also optimal for the follower problem (2.5) given the leader choice  $x$  and consequently  $(x, y)$  is feasible to (2.4) and has a better objective value than  $(x^*, y^*)$ . That is a contradiction to  $(x^*, y^*)$  being optimal for (2.4). Hence  $(x^*, y^*)$  is a local optimal solution to (3.4).  $\square$

Theorem 3.9 does not yield global optimality in the modified problem, even for linear bilevel problems, as we can see in the following example:

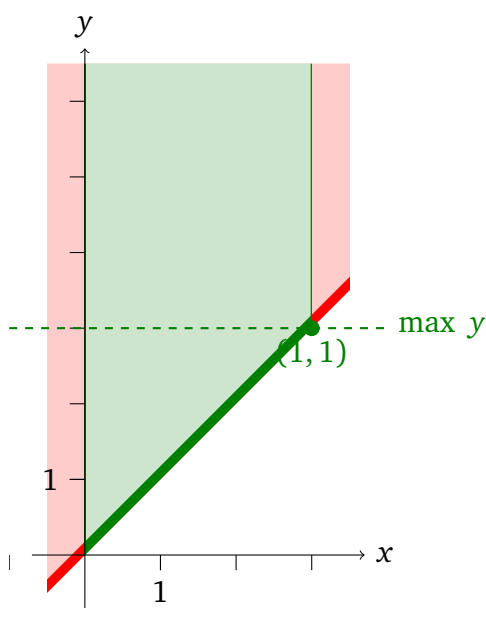
**Example 3.10.** *The linear bilevel program*

$$\max_{x,y} x \quad (3.10a)$$

$$s. t. \quad 0 \leq x \leq 1, \quad (3.10b)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (3.10c)$$

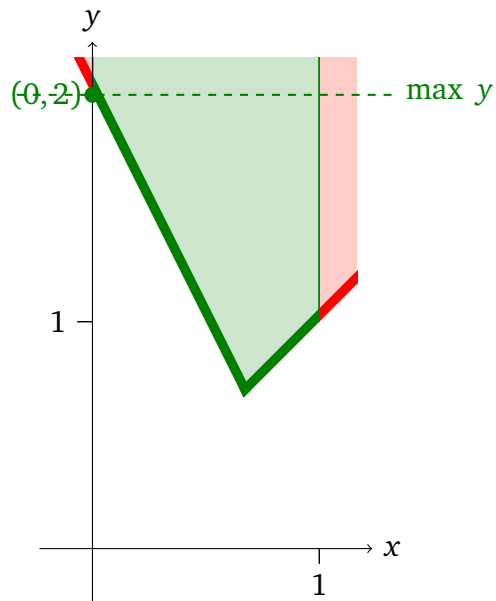
$$s. t. \quad \bar{y} \geq x. \quad (3.10d)$$



has global optimum  $(1, 1)$  with objective value 1. The following modified problem that adds  $2x + y \geq 2$  to the follower

$$\begin{aligned} \max_x \quad & y \\ s.t. \quad & 0 \leq x \leq 1, \\ & \min_y y \\ & s.t. \quad x \leq y. \\ & 2x + y \geq 2. \end{aligned}$$





has global optimum  $(0, 2)$  with objective value 2. However,  $2 \cdot 1 + 1 > 2$  is satisfied.

---

## 4 Theoretical Foundation for Presolve of Linear Bilevel Programs

In this chapter we present several theorems we have derived, which justify certain modifications of linear bilevel programs in a sense that we show that these modifications do not change the optimal solution.

**Remark 4.1.** *The linear bilevel program (2.4) can be written as*

$$\max_{(x,y) \in IR} c^\top x + d^\top y$$

and the inducible region  $IR$  of (2.4) can be written as

$$IR = \Omega_L \cap \bigcup_{x \in \mathbb{R}^n} \left( \{x\} \times \arg \min_y \{e^\top y \mid (x,y) \in \Omega_F\} \right) \quad (4.1)$$

$$= \Omega \cap \bigcup_{x \in \mathbb{R}^n} \left( \{x\} \times \arg \min_y \{e^\top y \mid (x,y) \in \Omega_F\} \right). \quad (4.2)$$

Equation (4.1) shows that the follower constraints indeed only contribute to the linear bilevel program (2.4) through the polyhedron  $\Omega_F$  they describe and (4.2) shows that the leader constraints only contribute through the polyhedron  $\Omega$  they describe together with the follower constraints.

This remark therefore justifies changes to the leader or follower constraints that do not change the respective polyhedrons, which in return justifies numerous presolve techniques applied to these constraints/polyhedrons.

**Corollary 4.2.** *Any changes applied to the follower constraints of (2.4) that do not change the follower polyhedron, will not change the inducible region of (2.4).*

*Proof.* This corollary directly derives from Remark 4.1. □

Next up we present a small but essential theorem on the removal of irrelevant constraints. For this, we define the modified follower constraints

$$C_{\neq i}x + D_{\neq i}y \leq b_{\neq i},$$

that are missing the  $i$ -th row, which is the constraint  $C_i x + D_i y \geq b_i$ . Furthermore, let

$$\Psi_{\neq i}(x) := \arg \min_{y \in \mathbb{R}^m} \{e^\top y \mid D_{\neq i}y \geq b_{\neq i} - C_{\neq i}x\}$$

be the rational reaction set of said modified follower problem. Lastly,

$$\max_{x,y} c^\top x + d^\top y \quad (4.3a)$$

$$\text{s. t. } Ax + By \leq a, \quad (4.3b)$$

$$y \in \Psi_{\neq i}(x), \quad (4.3c)$$

is the modified bilevel program that misses the  $i$ -th follower constraint and  $IR_{\neq i}$  is its inducible region.

---

**Corollary 4.3.** Let  $i \in \{1, \dots, l\}$  be a follower constraint of (2.4d) such that

$$\{(x, y) \mid Cx + Dy \geq b\} = \{(x, y) \mid C_{\neq i}x + D_{\neq i}y \geq b_{\neq i}\}.$$

In that sense, the  $i$ -th constraint is irrelevant for the description of the follower polyhedron. Then (2.4) and (4.3) have the same inducible region and therefore the same optimal solutions.

*Proof.* This corollary directly derives from Remark 4.1. □

This theorem gives us a simple criterion for the removal of follower constraints. Analogously, we define the program

$$\max_{x,y} \quad c^\top x + d^\top y \tag{4.4a}$$

$$\text{s. t.} \quad A_{\neq i}x + B_{\neq i}y \leq a_{\neq i}, \tag{4.4b}$$

$$y \in \Psi(x), \tag{4.4c}$$

where the  $i$ -th leader constraint is missing and derive the following criterion for the removal of leader constraints:

**Corollary 4.4.** Let  $i \in \{1, \dots, k\}$  be a leader constraint of (2.4b) such that

$$\{(x, y) \mid Ax + By \leq a, Cx + Dy \geq b\} = \{(x, y) \mid A_{\neq i}x + B_{\neq i}y \leq a_{\neq i}, Cx + Dy \geq b\},$$

which means that the high point relaxations of (2.4) and (4.4) have the same feasible sets. Then (2.4) and (4.4) have the same inducible region and therefore the same optimal solutions.

*Proof.* This corollary directly derives from Remark 4.1. □

## 4.1 Independence of Inactive Constraints for Linear Programs

We will now present a concept similar to the independence of irrelevant constraints that is also taken for granted in linear programs, which we call *independence of inactive constraints*.

**Example 4.5.** Consider the linear program

$$\min_{x,y} \quad y \tag{4.5a}$$

$$\text{s. t.} \quad y \geq x, \tag{4.5b}$$

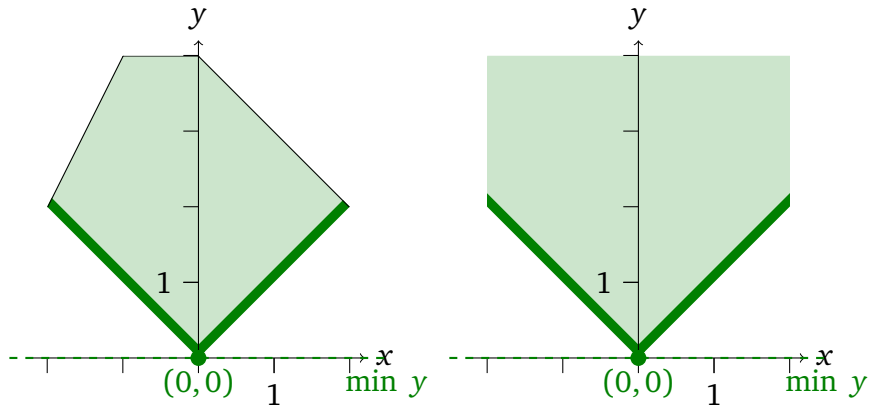
$$y \geq -x, \tag{4.5c}$$

$$y \leq 2, \tag{4.5d}$$

$$y \leq 2x + 3, \tag{4.5e}$$

$$y \leq -x + 2. \tag{4.5f}$$

This program has the optimal solution  $(0, 0)$ . The active constraints for that solution are (4.5b) and (4.5c). It is easy to see that we can remove all the inactive constraints and the optimal solution remains the same.



This independence of inactive constraints holds true for all linear programs, or generally for all optimization problems where local optimality implies global optimality.

However, linear bilevel programs are generally not convex, which is why local optimality does not imply global optimality as we saw in Example 2.1. Once again, the following examples illustrate that linear bilevel programs generally are not independent of inactive constraints.

**Example 4.6.** *The linear bilevel program*

$$\min_{x,y} y \quad (4.6a)$$

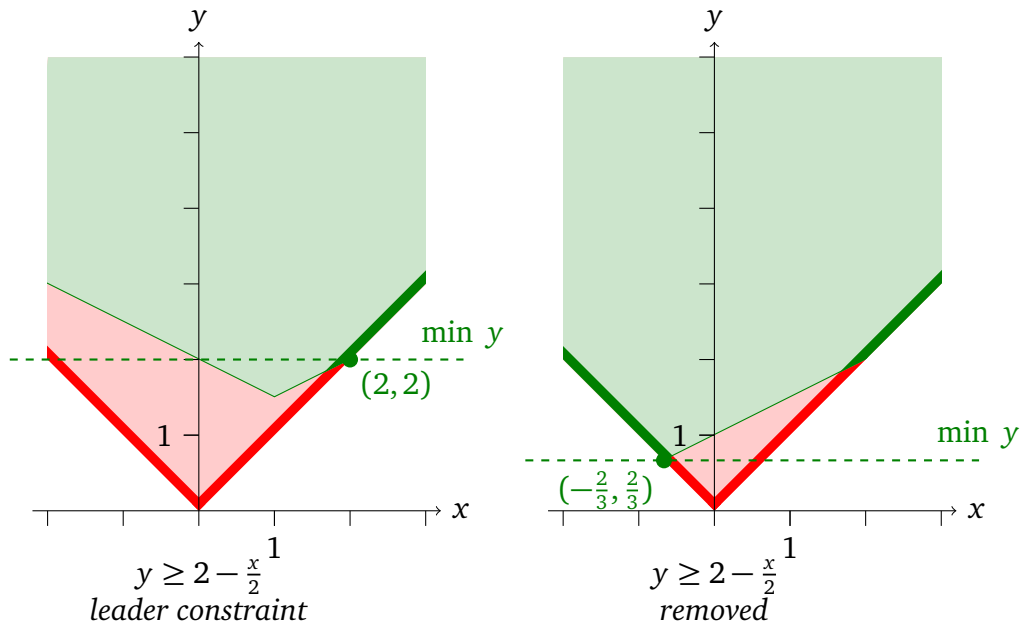
$$s. t. \quad y \geq 1 + \frac{x}{2}, \quad (4.6b)$$

$$y \geq 2 - \frac{x}{2}, \quad (4.6c)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (4.6d)$$

$$s. t. \quad \bar{y} \geq x, \quad (4.6e)$$

$$\bar{y} \geq -x. \quad (4.6f)$$



has the optimal solution  $(2, 2)$ , for which the leader constraint  $y \geq 2 - \frac{x}{2}$  is inactive. However, removing this constraint will result in the better optimal solution  $(-\frac{2}{3}, \frac{2}{3})$ .

**Example 4.7.** The linear bilevel program

$$\min_{x,y} x - y \tag{4.7a}$$

$$\text{s. t. } y \leq \frac{5}{2} + \frac{3}{2}x, \tag{4.7b}$$

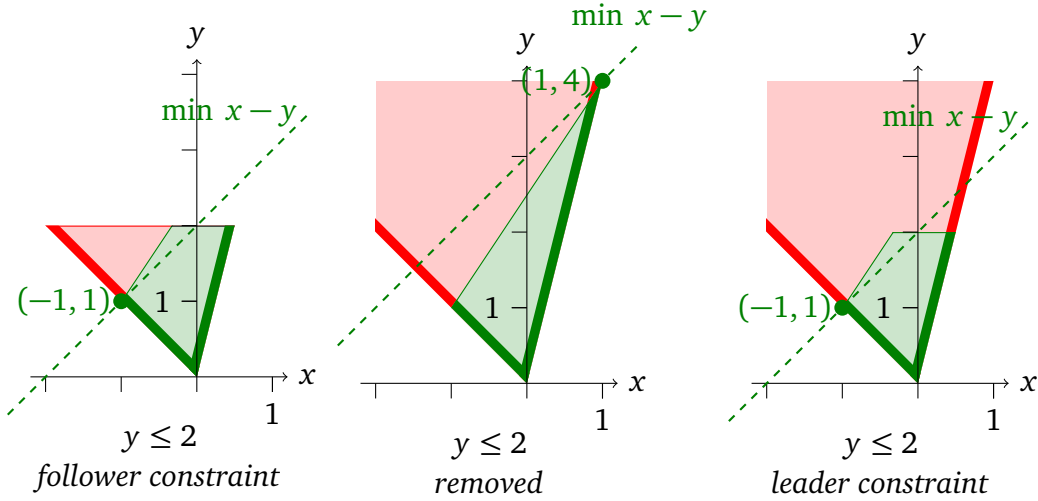
$$y \in \arg \min_{\bar{y}} \bar{y} \tag{4.7c}$$

$$\text{s. t. } \bar{y} \geq 4x, \tag{4.7d}$$

$$\bar{y} \geq -x, \tag{4.7e}$$

$$\bar{y} \leq 2. \tag{4.7f}$$

has the optimal solution  $(-1, 1)$ , for which the follower constraint  $y \leq 2$  is inactive. Removing this constraint will result in the better optimal solution  $(1, 4)$ .



**Observation 4.8.** *It is noteworthy that the constraint  $y \leq 2$  in Example 4.7 could be a leader constraint instead of a follower constraint and the solution to (4.7) would remain the same. This arouses the suspicion that inactive follower constraints could be moved to the leader, which we will investigate in Section 4.4.*

## 4.2 Consequences of Removing Inactive Constraints and Adding Irrelevant Constraints

In this section we investigate the consequences of removing inactive constraints from a linear bilevel program. For this we consider the general linear bilevel program  $P$  given by (2.4). Let  $P_L(i)$  be the modification of  $P$  where the  $i$ -th leader constraint has been removed given by

$P_L(i)$ :

$$\max_{x,y} c^\top x + d^\top y \quad (4.8a)$$

$$\text{s. t. } A_{\neq i}x + B_{\neq i}y \leq a_{\neq i}, \quad (4.8b)$$

$$y \in \arg \min_{\bar{y}} e^\top \bar{y} \quad (4.8c)$$

$$\text{s. t. } D\bar{y} \geq b - Cx. \quad (4.8d)$$

and  $P_F(j)$  the modification of  $P$  where the  $j$ -th follower constraint has been removed given by

$P_F(j)$ :

$$\max_{x,y} c^\top x + d^\top y \quad (4.9a)$$

$$\text{s. t. } Ax + By \leq a, \quad (4.9b)$$

$$y \in \arg \min_{\bar{y}} e^\top \bar{y} \quad (4.9c)$$

$$\text{s. t. } D_{\neq j}\bar{y} \geq b_{\neq j} - C_{\neq j}x. \quad (4.9d)$$

---

Furthermore, let  $IR(P)$ ,  $IR(P_L(i))$  and  $IR(P_F(j))$  be the inducible regions and  $Opt(P)$ ,  $Opt(P_L(i))$  and  $Opt(P_F(j))$  be the sets of optimal solutions of the three programs.

Example 4.7 has shown that linear bilevel programs are generally not independent of inactive constraints. However, the following theorem states that, when removing a follower constraint that is inactive for an optimal solution to the linear bilevel program that solution remains bilevel feasible.

**Theorem 4.9.** *Let  $(x^*, y^*) \in IR(P)$  and  $j \in \{1, \dots, l\}$  with*

$$C_j x^* + D_j y^* > b_j.$$

*Then  $(x^*, y^*) \in IR(P_F(j))$ .*

*Proof.* Consider the follower program of  $P$  for  $x = x^*$ :

$$\min_y e^\top y \tag{4.10a}$$

$$\text{s. t. } Dy \geq b - Cx^*. \tag{4.10b}$$

By requirement,  $y^*$  is an optimal solution to this problem and  $C_j x + D_j y \geq b_j$  is an inactive constraint for  $y^*$ . Theory on linear programs now tells us that  $y^*$  is also optimal for

$$\min_y e^\top y \tag{4.11a}$$

$$\text{s. t. } D_{\neq j} y \geq b_{\neq j} - C_{\neq j} x^*. \tag{4.11b}$$

Therefore,  $y^*$  is optimal to the follower problem of  $P_F(j)$  under the choice of  $x^*$ . Since the leader constraints of  $P$  and  $P_F(j)$  are the same,  $(x^*, y^*)$  is also bilevel feasible. Hence,  $(x^*, y^*) \in IR(P_F(j))$ .  $\square$

This tells us that removing inactive follower constraints preserves bilevel feasibility. Obviously, removing any leader constraint (not just inactive ones) will also preserve bilevel feasibility:

**Corollary 4.10.** *Let  $(x^*, y^*) \in IR(P)$  and  $i \in \{1, \dots, k\}$ . Then  $(x^*, y^*) \in IR(P_L(i))$ .*

The following example shows that removing active follower constraints generally does not preserve bilevel feasibility:

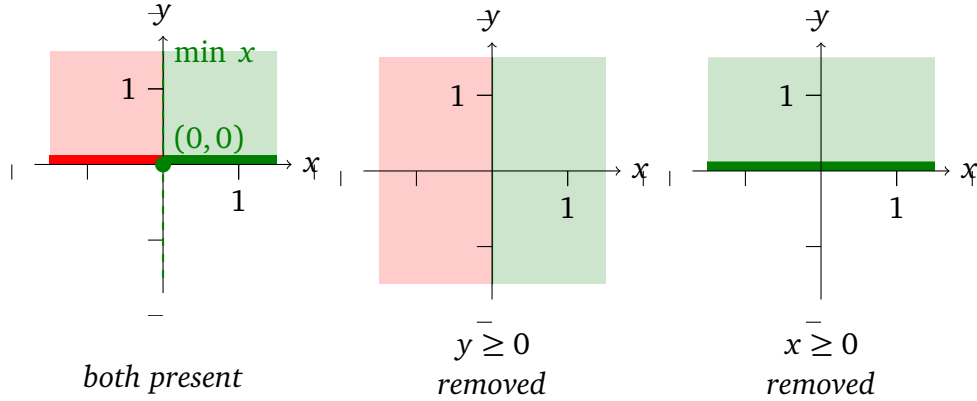
**Example 4.11.** *Consider the linear bilevel program*

$$\min_{x,y} x \tag{4.12a}$$

$$\text{s. t. } x \geq 0, \tag{4.12b}$$

$$y \in \arg \min_{\bar{y}} \bar{y} \tag{4.12c}$$

$$\text{s. t. } \bar{y} \geq 0. \tag{4.12d}$$



All  $(x^*, 0)$  with  $x^* \geq 0$  are bilevel feasible and have  $y \geq 0$  as an active constraint. Removing that constraint makes the follower problem unbounded and the whole program infeasible. However, removing the constraint  $x \geq 0$ , which is active for the optimal solution  $(0, 0)$  preserves the feasibility of  $(0, 0)$ , it just makes the bilevel program unbounded by adding new feasible solutions.

For the sake of completeness, we can derive a theorem similar to Theorem 4.9 but for adding irrelevant constraints.

**Theorem 4.12.** Let  $(x^*, y^*) \in \text{Opt}(P)$  and  $u^\top x + v^\top y \geq w$  be an arbitrary new constraint with

$$u^\top x^* + v^\top y^* \geq w.$$

When modifying  $P$  by adding the constraint  $u^\top x + v^\top y \geq w$  as a follower constraint,  $(x^*, y^*)$  will still be a bilevel feasible solution. Therefore, the optimal objective value of the modified  $P$  is either the same or better.

*Proof.* The new follower problem with  $x = x^*$  is

$$\min_y e^\top y \quad (4.13a)$$

$$\text{s. t. } Dy \geq b - Cx^*, \quad (4.13b)$$

$$v^\top y \geq w - u^\top x^*. \quad (4.13c)$$

Since  $y^*$  must still be feasible, it also is optimal for this problem. Furthermore, the new linear bilevel program

$$\max_{x,y} c^\top x + d^\top y \quad (4.14a)$$

$$\text{s. t. } Ax + By \leq a, \quad (4.14b)$$

$$y \in \arg \min_{\bar{y}} e^\top \bar{y} \quad (4.14c)$$

$$\text{s. t. } D\bar{y} \geq b - Cx, \quad (4.14d)$$

$$v^\top \bar{y} \geq w - u^\top x. \quad (4.14e)$$

has the same leader constraint. Hence,  $(x^*, y^*)$  must still be bilevel feasible.  $\square$



---

While we saw that we can neither easily add irrelevant constraints nor remove inactive constraints, we at least know that in either case the old optimal solution will still be feasible.

### 4.3 Independence of Inactive Constraints for Linear Bilevel Programs

Analogously to Definition 3.2 we define

**Definition 4.13** (Independence of inactive constraints). *P is independent of inactive constraints if for all  $i \in \{1, \dots, k\}$  and  $j \in \{1, \dots, l\}$*

$$(x^*, y^*) \in \text{Opt}(P_L(i)) \quad \forall (x^*, y^*) \in \text{Opt}(P) : A_i x^* + B_i y^* < a_i$$

and

$$(x^*, y^*) \in \text{Opt}(P_F(j)) \quad \forall (x^*, y^*) \in \text{Opt}(P) : C_j x^* + D_j y^* > b_j$$

hold.

*This means that a linear bilevel program is independent of inactive constraints if, after modifying the program by removing a constraint, every optimal solution of the original program that strictly satisfies the arbitrary constraint is also an optimal solution of the modified program.*

We now introduce the following lemma that we use to derive a connection between independence of irrelevant constraints and independence of inactive constraints.

**Lemma 4.14.** *If (4.6) is IIC, then every optimal solution to (4.6) is an optimal solution of its high point relaxation.*

*Proof.* If (4.6) is not degenerate, then we know by Theorem 3.4 that there is an optimal solution  $(x^*, y^*)$  to (4.6) that is also an optimal solution to its high point relaxation. Since (4.6) and its high point relaxation have the same objective function, all optimal solution to (4.6) must be optimal solutions to its high point relaxation.

If (4.6) is degenerate either both the program and its high point relaxation are infeasible (in which case the lemma trivially holds) or

$$e = 0,$$

in which case (4.6) is equivalent to its high point relaxation and the lemma trivially holds, again.  $\square$

We can now derive that independence of irrelevant constraints is stronger than independence of inactive constraints:

**Theorem 4.15.** *If (4.6) is independent of irrelevant constraints then it is also independent of inactive constraints.*

---

*Proof.* Let  $(x^*, y^*) \in \text{Opt}(P)$ ,  $j = 1, \dots, l$  and  $C_j x^* + D_j y^* > b_j$ . Since  $\text{Opt}(P) \subset \text{IR}(P)$  we know by Theorem 4.9 that  $(x^*, y^*) \in \text{IR}(P_F(j))$ . Furthermore, we know by Lemma 4.14 that  $(x^*, y^*)$  is optimal for the high point relaxation of  $P$  given by (2.7). The high point relaxation of  $P_F(j)$  is given by

$$\max_{x,y} c^\top x + d^\top y \quad (4.15a)$$

$$\text{s. t. } Ax + By \leq a, \quad (4.15b)$$

$$C_{\neq j} x + D_{\neq j} y \geq b_{\neq j}. \quad (4.15c)$$

Since  $C_j x^* + D_j y^* > b_j$  holds,  $(x^*, y^*)$  must also be optimal for (4.15). As  $(x^*, y^*)$  is bilevel feasible for  $P_F(j)$  and optimal for its high point relaxation, it must be optimal for  $P_F(j)$ . Hence,  $(x^*, y^*) \in \text{Opt}(P_F(j))$ .

For  $i \in \{1, \dots, k\}$  and  $A_i x^* + B_i y^* < a_i$  the proof goes analogously with Corollary 4.10 instead of Theorem 4.9.  $\square$

**Example 4.16.** *Now we will show that the inverse implication of Theorem 4.15 is generally wrong by considering the linear bilevel program*

$$\max_{x,y} y \quad (4.16a)$$

$$\text{s. t. } x \leq 1, \quad (4.16b)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (4.16c)$$

$$\text{s. t. } \bar{y} \leq x, \quad (4.16d)$$

$$\bar{y} \geq -x. \quad (4.16e)$$

*The only optimal solution to this program is  $(0, 0)$ , for which the only inactive constraint is  $x \leq 1$ . Removing the constraint still results in the same solution. Hence, the program is independent of inactive constraints. However, the solution to the high point relaxation of the problem is  $(1, 1)$  and not  $(0, 0)$ . Since the program is also not degenerate, it cannot be IIC according to Theorem 3.4. Specifically, the new follower constraint  $\bar{y} \geq x$  that is satisfied by  $(0, 0)$ , would produce the better bilevel optimal solution  $(1, 1)$ .*

In conclusion, independence of irrelevant constraints is a stronger property than independence of inactive constraints.

## 4.4 Moving Inactive Constraints from the Follower to the Leader

While we have seen that we cannot remove inactive constraints, we will now show that inactive follower constraints can be moved to the leader. For this we consider the linear bilevel program  $P$  given by (2.4). Let  $\tilde{P}(i)$  be the modified version of  $P$  where

---

the  $j$ -th follower constraint has been moved to the leader.  $\tilde{P}(i)$  is given by

$$\max_{x,y} \quad c^\top x + d^\top y \quad (4.17a)$$

$$\text{s. t.} \quad Ax + By \leq a, \quad (4.17b)$$

$$C_j x + D_j y \geq b_j, \quad (4.17c)$$

$$y \in \arg \min_{\bar{y}} \quad e^\top \bar{y} \quad (4.17d)$$

$$\text{s. t.} \quad D_{\neq j} \bar{y} \geq b_{\neq j} - C_{\neq j} x. \quad (4.17e)$$

While inactive constraints generally cannot be removed, they can actually be moved to the leader instead:

**Theorem 4.17.** *Let  $IR(P)$  and  $IR(\tilde{P}(j))$  be the inducible regions of  $P$  and  $\tilde{P}(i)$ .*

1. *If  $(x^*, y^*) \in IR(P)$  with  $C_j x^* + D_j y^* > b_j$ , then*

$$(x^*, y^*) \in IR(\tilde{P}(j)).$$

*This means that moving a constraint from the follower to the leader preserves the bilevel feasibility of points that strictly satisfy the constraint.*

2. *If  $(x^*, y^*) \in IR(\tilde{P}(j))$ , then*

$$(x^*, y^*) \in IR(P).$$

*This means that no new bilevel feasible points can occur because a constraint was moved to the leader.*

*Proof.* 1. Let  $P_F(j)$  be given as in (4.9). Then Theorem 4.9 implies

$$(x^*, y^*) \in IR(P_F(j)).$$

Furthermore, we obtain  $\tilde{P}$  by adding  $C_j x + D_j y \geq b_j$  to  $P_F(j)$  as a leader constraint. Therefore,

$$(x^*, y^*) \in IR(\tilde{P}(i)),$$

since  $(x^*, y^*)$  satisfies  $C_j x + D_j y \geq b_j$  by requirement.

2. The follower problem of  $P$  for  $x = x^*$  is

$$\min_y \quad e^\top y \quad (4.18a)$$

$$\text{s. t.} \quad Dy \geq b - Cx^*. \quad (4.18b)$$

and the follower problem of  $\tilde{P}(j)$  for  $x = x^*$  is

$$\min_y \quad e^\top y \quad (4.19a)$$

$$\text{s. t.} \quad D_{\neq j} y \geq b_{\neq j} - C_{\neq j} x^*. \quad (4.19b)$$

Since  $(x^*, y^*)$  is optimal for (4.19) by requirement and satisfies  $C_j x^* + D_j y^* \geq b_j$  (which is a leader constraint of  $\tilde{P}(j)$ , it is also optimal for (4.19). Furthermore,  $(x^*, y^*)$  also satisfies all leader constraints of  $P$  since they are a relaxation of the leader constraints of  $\tilde{P}(j)$ . Therefore,

$$(x^*, y^*) \in IR(P).$$

□

Theorem 4.17 directly implies that every follower constraint that is not active for optimal bilevel solutions can be moved to the leader while sustaining these optimal solutions. In that sense, every linear bilevel program is independent of whether inactive follower constraints are in the follower or the leader.

**Example 4.18.** *The linear bilevel program*

$$\max_{x,y} y \tag{4.20a}$$

$$\text{s. t. } x \leq \frac{5}{2}, \tag{4.20b}$$

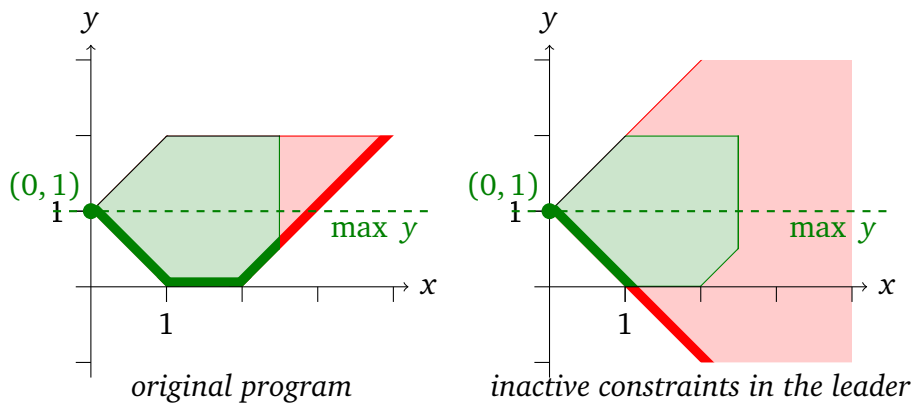
$$y \in \arg \min_{\bar{y}} \bar{y} \tag{4.20c}$$

$$\text{s. t. } \bar{y} \geq 1 - x, \tag{4.20d}$$

$$\bar{y} \geq -2 + x, \tag{4.20e}$$

$$\bar{y} \leq 1 + x, \tag{4.20f}$$

$$0 \leq \bar{y} \leq 2 \tag{4.20g}$$



has the optimal solution  $(0, 1)$  for which the follower constraint  $y \geq -2 + x$  and the bounds of  $y$  are inactive and can therefore be moved to the leader without changing the optimal solution. This example also shows that  $IR^*$  and  $IR$  are not generally the same in Theorem 4.17.

In linear programs, inactive constraints can be added or removed without changing the optimal solution. Therefore, one can remove any constraint of a linear program, solve the relaxed linear program, test if the solution violates the removed constraints and add them again to cut off the infeasible solution. This is called separation.

A similar approach for linear bilevel programs could be to start with all follower constraints as leader constraints and then successively move them to the follower as needed. However, this method will not work because moving inactive constraints from the follower to the leader *removes* suboptimal solutions while removing inactive constraints from a linear program adds suboptimal solutions. This illustrates that linear bilevel programs actually have *more* bilevel feasible solutions when more constraints are moved from the leader to the follower.

**Example 4.19.** Consider the program

$$\min_{x,y} \frac{x}{2} + y \quad (4.21a)$$

$$\text{s. t. } y \geq 1, \quad (4.21b)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (4.21c)$$

$$\text{s. t. } \bar{y} \geq x, \quad (4.21d)$$

$$\bar{y} \geq -x, \quad (4.21e)$$

which has the optimal solution  $(-1, 1)$ . However, starting with the constraint  $y \geq -x$  as a leader constraint yields

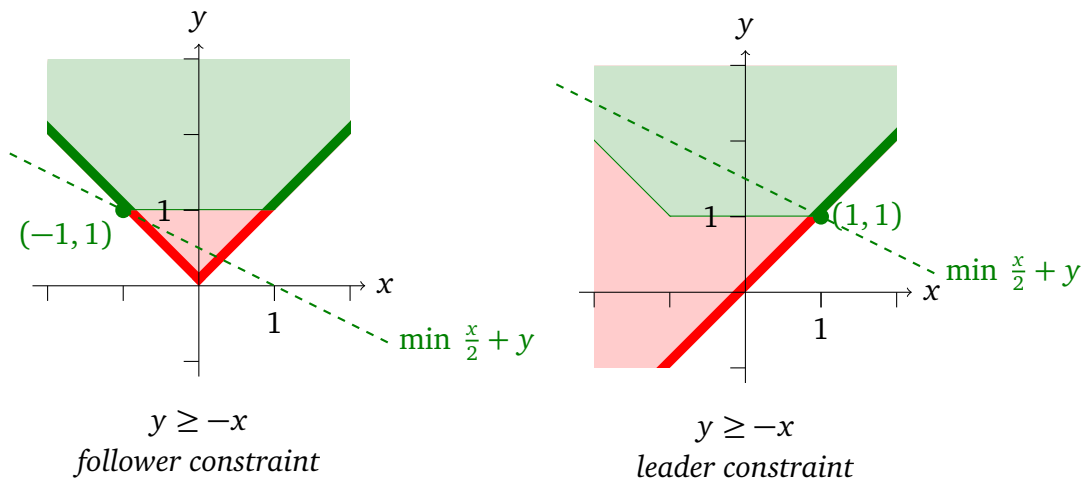
$$\min_{x,y} \frac{x}{2} + y \quad (4.22a)$$

$$\text{s. t. } y \geq 1, \quad (4.22b)$$

$$y \geq -x, \quad (4.22c)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (4.22d)$$

$$\text{s. t. } \bar{y} \geq x, \quad (4.22e)$$



---

*which has the optimal solution  $(1, 1)$ . This solution is also bilevel feasible in the original program and there is no direct way to tell that the original program that had  $y \geq -x$  in the follower, has a better solution, since the constraint  $y \geq -x$  is not even active for  $(1, 1)$ .*

To summarize this chapter, we defined the property independence of inactive constraints, which is similar to the classic IIC property as both are taken for granted in linear programs, but are not the same since IIC implies independence of inactive constraints but not vice versa. We also showed that both operations, namely adding irrelevant and removing inactive constraints, bear a certain monotony in the objective, because they both do not invalidate existing solutions but possibly introduce better ones. While we generally cannot expect to be able to remove inactive constraints, we at least saw that we can move these constraints to the leader. We will derive presolve techniques based on this idea in the next chapters.

---

## 5 Finding Inactive Constraints with the $K$ th-best Algorithm

While the  $K$ th-best algorithm can find a quick solution, it is generally a lot slower than, for example, solving the KKT reformulation with a MIP solver. However, we will now present a concept that shows how the  $K$ th-best algorithm can be used for presolve. Specifically, the idea is to terminate the  $K$ th-best when it does not find an optimal solution in an appropriate amount of time and then use the information on the visited vertices to find inactive constraints.

First, we present the algorithm itself. For this, we consider the linear bilevel program (2.4), its follower problem (2.5) and its high point relaxation (2.4).

Furthermore, we define

$$(x^*, y^*) \in \arg \max_{x, y} \{c^\top x + d^\top y \mid Ax + By \leq a, Cx + Dy \geq b\}$$

as an optimal solution of (2.7) and

$$G = (V, E)$$

as the undirected graph representing the basis structure of the combined polyhedron  $\Omega$ . This means that  $V$  are the extreme points of  $\Omega$  and  $\{v, w\} \in E$  if and only if  $v$  and  $w$  are adjacent in  $\Omega$  in the sense that their basis representations differ by only one basis variable. For convenience we define

$$\text{Adj}(v) = \{w \in V \mid \{v, w\} \in E\}$$

as the neighbours of  $v$  in  $G$  for any  $v \in V$  and

$$\text{Ver}(j) = \{(x, y) \in V \mid C_j x + D_j y = b_j\}$$

as the vertices that have  $j$  as an active constraint for any  $j \in \{1, \dots, l\}$ .

The  $K$ th-best algorithm, as first proposed in [5], is a solver for bilevel problems that makes use of the fact that there is always an optimal solution to the bilevel problem that is also an extreme point of  $\Omega$ . This was shown in [4].

The algorithm iterates over the extreme points of said feasible set starting with an optimal solution to the high point relaxation and then continues to the second best extreme point and so on. Each extreme point is tested for follower optimality and if it is follower optimal, it means that it is bilevel feasible and, due to the descending order in which the extreme points are traversed, also bilevel optimal.

The  $K$ th-best algorithm for the linear bilevel program (2.4) works like this:

---

---

**Algorithm 1** Kth-best algorithm

---

**Require:** (2.7) is solvable

$S$  is a sorted list of variable vectors  $(x, y)$ , sorted by  $c^T x + d^T y$  descending

$S \leftarrow \{(x^*, y^*)\}$

$i \leftarrow 1$

**while**  $i \leq |S|$  **do**

$(x_i, y_i) = S[i]$

**if**  $y_i$  is optimal for (2.5) with  $x = x_i$  **then**

**return**  $(x_i, y_i)$  is optimal for (2.4)

**else**

$S \leftarrow S \cup \text{Adj}(x_i, y_i)$  (keep  $S$  sorted)

$i \leftarrow i + 1$

**end if**

**end while**

**return** (2.4) is infeasible

---

**Example 5.1.** Consider the linear bilevel program

$$\max_{x, y \in \mathbb{R}} y \quad (5.1a)$$

$$s. t. \quad y \geq -2 + x, \quad (5.1b)$$

$$1 \leq x \leq 4, \quad (5.1c)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (5.1d)$$

$$s. t. \quad \bar{y} \leq \frac{3}{2} + x, \quad (5.1e)$$

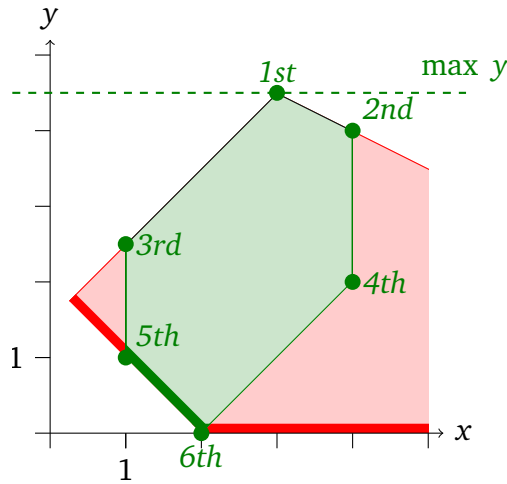
$$\bar{y} \leq 6 - \frac{x}{2} \quad (5.1f)$$

$$\bar{y} \geq 2 - x, \quad (5.1g)$$

$$\bar{y} \geq 0. \quad (5.1h)$$

The iteration order for the Kth-best applied to this program looks like this:





Note that the 6th node is never visited.

We will now present how the information gathered during the execution of Algorithm 1 can be used to find active constraints in the follower.

Let  $S$  and  $i$  be the respective variables of Algorithm 1 after an early termination. That means that the first  $i$  elements of  $S$  have been determined as not bilevel feasible. Accordingly, we split  $S$  into the bilevel infeasible points

$$P := \{S[1], \dots, S[i]\}$$

and the unchecked points

$$Q := \{S[i+1], \dots\}.$$

We now want to find follower constraints  $j \in \{1, \dots, l\}$  with

$$\text{Ver}(j) \subset P,$$

since these constraints do not contain any bilevel feasible vertices and are therefore inactive for all bilevel feasible solutions and can be moved to the leader according to Theorem 4.17. Rather than calculating  $\text{Ver}(j)$  for every  $j \in \{1, \dots, l\}$  we can deduce a better approach to find these inactive constraints.

**Theorem 5.2.** Let  $j \in \{1, \dots, l\}$  with

$$\text{Ver}(j) \cap Q = \emptyset \tag{5.2}$$

and

$$\text{Ver}(j) \cap P \neq \emptyset. \tag{5.3}$$

Then

$$\text{Ver}(j) \subset P.$$

---

*Proof.* Since all points in  $P$  were determined to be bilevel infeasible and subsequently their neighbors in  $G$  were added to  $S$ , we can deduce that

$$\text{Adj}(x, y) \subset S \quad \forall (x, y) \in P. \quad (5.4)$$

Aside from that, polyhedron theory tells us that the set of vertices  $\text{Ver}(j)$  is connected in  $G$  for any  $j \in \{1, \dots, l\}$ .

We now assume that

$$\text{Ver}(j) \not\subset P.$$

Since (5.3) holds and  $\text{Ver}(j)$  is connected in  $G$ , there must be a  $(x, y) \in \text{Ver}(j) \cap P$  which has a neighbor

$$(\tilde{x}, \tilde{y}) \in \text{Adj}(x, y) \setminus P.$$

Furthermore, (5.4) holds and we deduce that

$$(\tilde{x}, \tilde{y}) \in Q,$$

which implies

$$\text{Ver}(j) \cap Q \neq \emptyset,$$

in contradiction to (5.2). Consequently,

$$\text{Ver}(j) \subset P.$$

□

Theorem 5.2 gives a criterion for follower constraints to be inactive. The requirements for theorem 5.2 are easily confirmed by testing  $C_j x + D_j y = b_j$  for every  $(x, y) \in S$ . On top of that, basis information on the nodes that was used to traverse the graph  $G$  can be used to reduce the amount of constraints the node has to be tested against.

**Example 5.3.** We illustrate the  $S$ ,  $P$  and  $Q$  at every iteration of Algorithm 1:

$i$	$S$	$P$	$Q$
start	{1}	$\emptyset$	{1}
after 1	{1, 2, 3}	{1}	{2, 3}
after 2	{1, 2, 3, 4}	{1, 2}	{3, 4}
after 3	{1, 2, 3, 4, 5}	{1, 2, 3}	{4, 5}
after 4	{1, 2, 3, 4, 5, 6}	{1, 2, 3, 4}	{5, 6}
after 5	{1, 2, 3, 4, 5, 6}	{1, 2, 3, 4, 5}	{6}

When we apply Theorem 5.2 to Example 5.1, visiting node 2nd determines that the constraint  $y \geq 6 - \frac{x}{2}$  is inactive, visiting node 3rd determines that  $y \leq \frac{3}{2} + x$  is inactive and visiting node 4th determines that  $1 \leq 4$  is inactive (which already is a leader constraint, however).

The  $K$ th-best algorithm was implemented in the course of this thesis involving around 2000 lines of code including two different approaches for traversing the extreme points. The latest state of the code is stored in Martin Schmidt's GitLab repository in Trier under the GitHash ffe75810bf32f36e76d43423455ba19c9125fdac.

---

## 6 Presolve Techniques for Linear Bilevel Programs

In this chapter we will go over several presolve techniques that are successfully employed in linear programming and mixed integer programming and will determine, if and how they can be used to presolve linear bilevel programs.

### 6.1 Bound Strengthening

This presolve technique, as explained in [1], tries to tighten variable bounds without cutting off any feasible points, so that they can be achieved by at least one feasible point instead of being unachievable.

#### 6.1.1 Bound Strengthening on Linear Programs

Before we investigate this technique in the context of linear bilevel programming, we will give a short explanation of the essentials by applying the method called *bound strengthening* as seen in [1] to a linear program.

**Example 6.1.** Consider the  $\mathbb{R}^2$ -polyhedron given by the constraints

$$y \geq 3 - \frac{x}{2}, \quad (6.1a)$$

$$y \leq 6 - 2x, \quad (6.1b)$$

$$y \geq 1, \quad (6.1c)$$

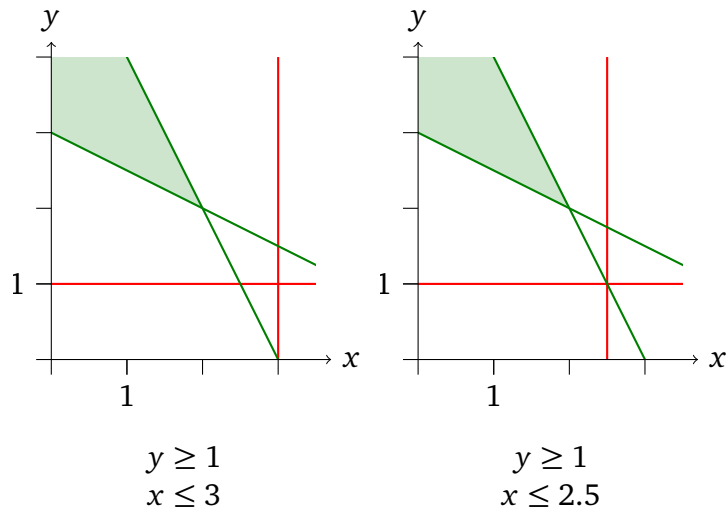
$$x \leq 3. \quad (6.1d)$$

By inserting  $y \geq 1$  into  $y \leq 6 - 2x$  one finds that

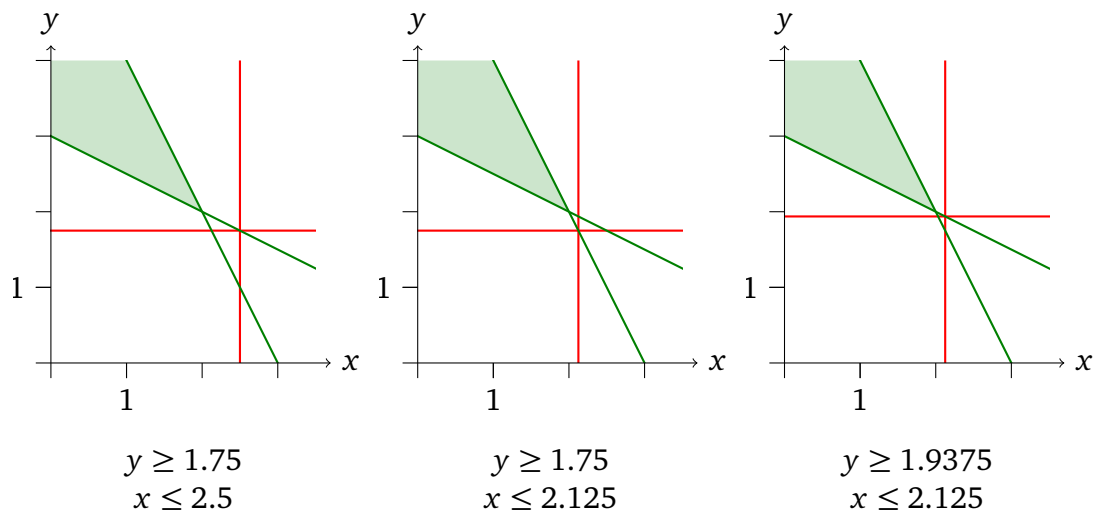
$$1 \leq 6 - 2x \quad (6.2)$$

$$\iff x \leq \frac{5}{2}. \quad (6.3)$$

The bound of  $x$  can now be tightened to  $x \leq \frac{5}{2}$ , which effectively moves the bound on  $x$  to the intersection of the constraint  $y \leq 6 - 2x$  and the bound  $y \geq 1$ :



The same procedure can now be applied to the constraint  $y \geq 3 - \frac{x}{2}$  and the bound  $x \leq \frac{5}{2}$  to tighten the bound  $y \geq 1$  to  $y \geq 1.75$ . Afterwards, the bound on  $x$  can be tightened even further using the new bound on  $y$  and so on:



Ultimately, this converges to the tight bounds  $x \leq 2$  and  $y \geq 2$ . Obviously, in this example the intersection point  $(2, 2)$  of the two constraints could have been calculated straight away, but that does not work in higher dimensions.

We generalize this technique to the linear program

$$\max_x c^\top x \tag{6.4a}$$

$$\text{s. t. } Ax \leq b, \tag{6.4b}$$

$$l \leq x \leq u. \tag{6.4c}$$

---

Let  $A_i x \leq b_i$  be a single constraint, let  $x_j$  be the variable whose bounds we want to strengthen, let  $A_{ij} \neq 0$  and  $S = \{1, \dots, n\} \setminus \{j\}$ . Then we rewrite the constraint as

$$A_{iS}x_S + A_{ij}x_j \leq b_i$$

and solve by  $x_j$  to get

$$x_j \leq (b_i - A_{iS}x_S)/A_{ij} \quad \text{if } A_{ij} > 0, \quad (6.5)$$

or

$$x_j \geq (b_i - A_{iS}x_S)/A_{ij} \quad \text{if } A_{ij} < 0. \quad (6.6)$$

In either case we can now estimate

$$l_{iS} := \sum_{\substack{p \in S \\ A_{ip} > 0}} A_{ip}l_p + \sum_{\substack{p \in S \\ A_{ip} < 0}} A_{ip}u_p \leq A_{iS}x_S.$$

$l_{iS}$  is also called the *minimal activity* of  $A_{iS}x_S$ . If we apply this to (6.5) and (6.6) we get

$$x_j \leq (b_i - l_{iS})/A_{ij} \quad \text{if } A_{ij} > 0,$$

or

$$x_j \geq (b_i - l_{iS})/A_{ij} \quad \text{if } A_{ij} < 0$$

which leads to the new bounds

$$u'_j := \min\{u_j, (b_i - l_{iS})/A_{ij}\} \quad \text{if } A_{ij} > 0,$$

or

$$l'_j := \max\{l_j, (b_i - l_{iS})/A_{ij}\} \quad \text{if } A_{ij} < 0.$$

We call this procedure *strengthening the bound  $u_j$  or  $(l_j)$  using the constraint  $A_i x \leq b_i$* .

As seen in Example 6.1, this scheme can be applied in an iterative fashion where one alternates between the bounds of different variables. In some cases, this process would need to be repeated infinitely in order for the bounds to converge, which we saw in Example 6.1. In [1] it is covered in detail how this infinite conversion is avoided. The basic idea is to stop iterating when the bounds improve too little within each step.

### 6.1.2 Bound Strengthening on Linear Bilevel Programs

We will now put bound strengthening in the context of linear bilevel programming. From a theoretical point of view a single bound strengthening step can be seen as adding a new constraint (the improved bound) while afterwards removing an obsolete

---

constraint (the old bound). We consider the linear bilevel program

$$\max_{x,y} c^\top x + d^\top y \quad (6.7a)$$

$$\text{s. t. } Ax + By \leq a, \quad (6.7b)$$

$$L \leq x \leq U, \quad (6.7c)$$

$$l \leq y \leq u, \quad (6.7d)$$

$$y \in \arg \min_{\bar{y}} e^\top \bar{y} \quad (6.7e)$$

$$\text{s. t. } D\bar{y} \geq b - Cx \quad (6.7f)$$

$$\bar{l} \leq \bar{y} \leq \bar{u}. \quad (6.7g)$$

with explicit bounds. Note that it can have bounds for  $y$  in the leader as well as in the follower.

First we present a theorem that we can strengthen variable bounds appearing in the leader without any restriction.

**Theorem 6.2.** *Strengthening the bounds (6.7c) and (6.7d) using any constraint of (6.7) will not change the inducible region  $IR$  of (6.7).*

*Proof.* We consider the combined polyhedron

$$\Omega = \{(x, y) \mid Ax + By \leq a, Cx + Dy \geq b, L \leq x \leq U, l \leq y \leq u, \bar{l} \leq y \leq \bar{u}\},$$

which is the domain of the high point relaxation of (6.7). We now apply bound strengthening to the bounds (6.7c) and (6.7d) using any constraint describing  $\Omega$ , which is any constraint of (6.7). This bound strengthening technique as it is described in [1] leaves  $\Omega$  unchanged. By Remark 4.1 the inducible region  $IR$  remains the same.  $\square$

We will now show which constraints can be used to strengthen follower bounds and which cannot be used.

**Theorem 6.3.** *Strengthening the bounds (6.7g) using any constraint in (6.7f) will not change the inducible region of 6.7.*

*Proof.* We consider the follower polyhedron

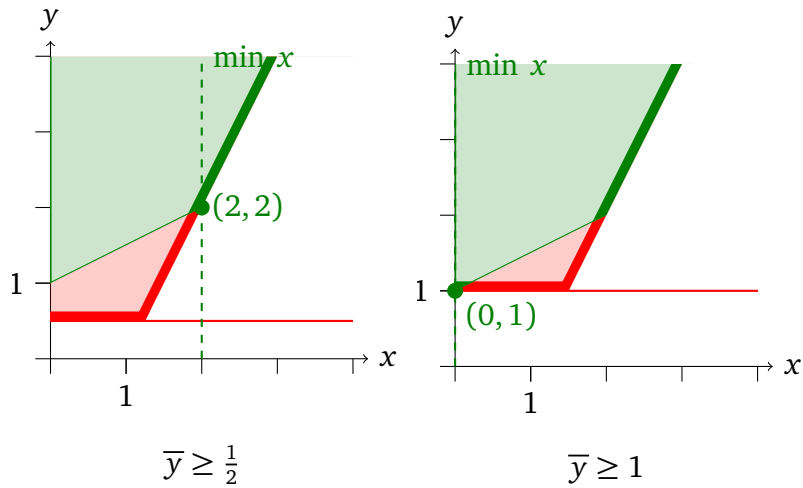
$$\Omega_F = \{(x, y) \mid Cx + Dy \geq b, \bar{l} \leq y \leq \bar{u}\}$$

We can now apply bound strengthening to the constraints and bounds describing  $\Omega_F$ . Since the bound strengthening technique as it is described in [1] leaves  $\Omega_F$  unchanged, the inducible region  $IR$  remains the same by Remark 4.1.  $\square$

While using follower constraints to strengthen follower bounds works fine, we now present an example where using leader constraints to strengthen follower bounds changes the optimal solution.

**Example 6.4.** *The linear bilevel program*

$$\begin{aligned}
 \min_{x,y \in \mathbb{R}} \quad & x \\
 \text{s. t.} \quad & y \geq \frac{x}{2} + 1, \\
 & x \geq 0, \\
 & y \in \arg \min_{\bar{y}} \bar{y} \\
 & \text{s. t.} \quad \bar{y} \geq 2x - 2, \\
 & \bar{y} \geq \frac{1}{2}
 \end{aligned}$$



has the optimal solution  $(2, 2)$ . When strengthening the bound  $\bar{y} \geq \frac{1}{2}$  using the constraint  $y \geq x/2 + 1$ , one finds that the minimal activity of  $x/2 + 1$  is 1 due to  $x \geq 0$ , which raises the bound of  $\bar{y}$  to  $\bar{y} \geq 1$ . This leads to the program

$$\begin{aligned}
 \min_{x,y \in \mathbb{R}} \quad & x \\
 \text{s. t.} \quad & y \geq \frac{x}{2} + 1, \\
 & x \geq 0, \\
 & y \in \arg \min_{\bar{y}} \bar{y} \\
 & \text{s. t.} \quad \bar{y} \geq 2x - 2, \\
 & \bar{y} \geq 1
 \end{aligned}$$

with the optimal solution  $(0, 1)$ .

---

## 6.2 Substitution

In this section we will investigate the possibility of substituting variables by equations. Specifically, if you have an equation  $a^\top x = b$  in a linear program then you can solve it for any  $x_i$  where  $a_i \neq 0$  and then replace any occurrence of  $x_i$ . Generally, while reducing the dimension of the problem, it also introduces a lot of new non-zero coefficients, which generally outweighs the benefits of the reduced dimensions. However, in certain cases it can be beneficial.

We now want to determine in which cases it is possible to substitute a variable in a linear bilevel program by an equation without changing the optimal solution. The rule for that is simple:

**Remark 6.5.** *If the equation used for substitution is a follower constraint, then any one variable (leader or follower), whose coefficient within said equation is not zero, can be substituted by that equation. This is done in two steps:*

1. *Replace every occurrence of the variable by the equation. This operation is justified by (4.1).*
2. *Remove the variable, which now does not appear in the program at all, from the program.*

*If the equation used for substitution is a leader constraint, then any one leader variable, whose coefficient within said equation is not zero, can be substituted by that equation. We justify this the same way as we did above.*

However, substituting a follower variable by an equation that is a leader constraint basically means to first introduce that equation as a follower constraint, which generally cannot be done without changing the optimal solution. The following example illustrates that:

**Example 6.6.** *The linear bilevel program*

$$\max_{x,y} y \quad (6.8a)$$

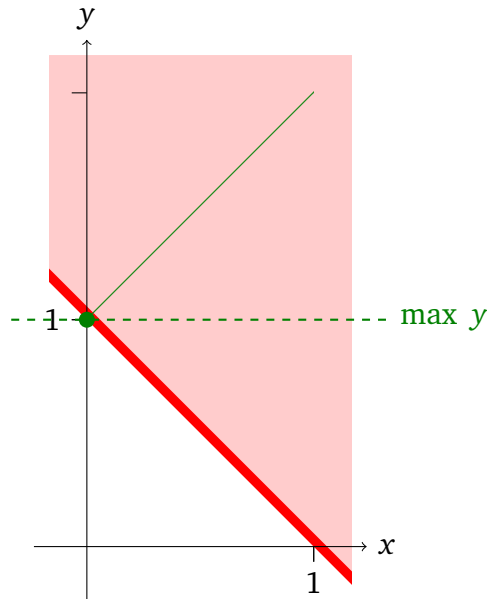
$$\text{s. t. } y = x + 1, \quad (6.8b)$$

$$0 \leq x \leq 1, \quad (6.8c)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (6.8d)$$

$$\text{s. t. } \bar{y} \geq 1 - x. \quad (6.8e)$$





has the optimal solution  $(0, 1)$ . When we substitute the follower variable  $y$  by the leader constraint  $y = x + 1$  we get the linear program

$$\max_x \quad x + 1 \tag{6.9a}$$

$$\text{s. t.} \quad 0 \leq x \leq 1, \tag{6.9b}$$

which has the optimal solution 1. Substituting back, we get the two-dimensional solution  $(1, 2)$ .

### 6.3 Duality Fixing

Another important presolve technique for single level programs is *duality fixing* as seen in [1] for MIPs and in [14] for bilevel programs. Consider the following linear program:

$$\begin{aligned} \min \quad & x_1 - x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 2, \\ & 2x_1 - x_2 \leq 1, \\ & x_1, x_2 \geq 0. \end{aligned}$$

From the perspective of optimization we want to minimize  $x_1$ . This also applies to the two constraints. If we did not have a lower bound for  $x_1$ , this would show that the problem was unbounded. However, the lower bound  $x_1 \geq 0$  prevents us from making  $x_1$  as small as we want and thus we fix  $x_1 = 0$ . This fixing only removes non-optimal solutions, since for every feasible point  $(x_1, x_2)$  with  $x_1 > 0$  the point  $(0, x_2)$  is also feasible and has a better objective value.

---

For a general linear program of the form

$$\begin{aligned} \max \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b, \end{aligned}$$

where the bounds (if any exist) are included in  $Ax \leq b$ , this can be generalized as follows:

Find a variable where all constraint coefficients are inverse to the objective coefficient except for one constraint coefficient and use that odd one out to fix the variable. Precisely, let  $i \in \{1, \dots, k\}$  and  $j \in \{1, \dots, l\}$  so that

$$\begin{aligned} A_{ji} &> 0, \\ A_{pi} &\leq 0 \quad \forall p \neq j, \\ c_i &> 0 \end{aligned}$$

or

$$\begin{aligned} A_{ji} &< 0, \\ A_{pi} &\geq 0 \quad \forall p \neq j, \\ c_i &< 0. \end{aligned}$$

Then substitute

$$x_i = \frac{1}{A_{ji}} \left( b_i - \sum_{p \neq i} A_{jp} x_p \right).$$

Note that potential variable bounds are included in  $Ax \leq b$ . If row  $j$  is e.g. a lower bound  $x_i > l_i$ , then  $\frac{1}{A_{ji}} \left( b_i - \sum_{p \neq i} A_{jp} x_p \right)$  shrinks down to  $l_i$  since  $A_{ji} = 1$ ,  $b_i = l_i$  and  $A_{jp} = 0 \quad \forall p \neq i$ .

This effectively removes the variable from the problem, thus reducing its dimension. If a variable has only non-negative or only non-positive coefficients (which includes that it has no lower/upper bound), then the problem is unbounded.

Now we look at a similar bilevel example where one leader variable only has non-negative coefficients:

$$\begin{aligned} \min_x \quad & x - y, \\ \text{s.t.} \quad & 3x - y \leq 3, \\ & 0 \leq x, \\ & \min_y \quad y, \\ & \text{s.t.} \quad 2x - y \leq 0. \end{aligned}$$

However, the follower problem is

$$\begin{aligned} \min_y \quad & y, \\ \text{s.t.} \quad & y \geq 2x, \end{aligned}$$

---

which translates to

$$y = 2x.$$

Substituting  $y$  with this equation in the original problem yields the linear program

$$\begin{aligned} \min_x \quad & -x, \\ \text{s.t.} \quad & x \leq 3, \\ & 0 \leq x, \end{aligned}$$

where  $x$  does not have non-negative coefficients anymore. We see that the optimal solution to the original problem is  $(3, 6)$ , which means that fixing  $x$  to its lower bound was wrong in the first place. This is due to the fact that we did not consider the follower optimality constraint on  $y$ , which generally cannot be seen as a positive or negative constraint on  $y$  since it is non-convex.

However, we can use duality fixing to fix follower variables. For this we are going to consider linear bilevel program given by (2.4) and its follower problem (2.5). The following theorem shows that we can use duality fixing on the follower problem without any consideration of the leader variables, leader constraints or leader objective.

**Theorem 6.7.** *Let  $i$  and  $j$  such that either*

$$\begin{aligned} D_{ji} &> 0, \\ D_{pi} &\leq 0 \quad \forall p \neq j, \\ e_i &> 0 \end{aligned}$$

or

$$\begin{aligned} D_{ji} &< 0, \\ D_{pi} &\geq 0 \quad \forall p \neq j, \\ e_i &< 0. \end{aligned}$$

Then adding the constraint

$$y_i = \frac{1}{D_{ji}} \left( e_i - C_{j,x} - \sum_{p \neq i} D_{jp} y_p \right) \quad (6.10)$$

to the leader of (2.4) will not change the set of bilevel feasible points.

*Proof.* Without loss of generality we can assume the first case, i.e.

$$\begin{aligned} D_{ji} &> 0, \\ D_{pi} &\leq 0 \quad \forall p \neq j, \\ e_i &> 0. \end{aligned}$$

For the sake of convenience we define

$$l_i := \frac{1}{D_{ji}} \left( e_i - C_{j,x} - \sum_{p \neq i} D_{jp} y_p \right).$$

---

Furthermore, let  $y$  be feasible to (2.5) for some leader decision  $x$ , therefore

$$y_i \geq l_i,$$

which justifies the nomenclature of  $l_i$ .

Assume a point  $y$  that feasible to (2.5) for some leader choice  $x$  and that satisfies  $y_i > l_i$ . Then  $\hat{y}$  with

$$\hat{y}_k = \begin{cases} y_k, & \text{if } k \neq i \\ l_i, & \text{if } k = i \end{cases}$$

is also feasible to (2.5) since  $D_{ki} \geq 0 \forall k \neq j$ . Furthermore,  $\hat{y}$  has a better objective value in (2.5) than  $y$  since  $e_i > 0$ . Consequently,  $y$  cannot have been optimal.

Conversely, all optimal solutions  $y$  of (2.5) must satisfy (6.10). Since the bilevel feasible set only contains solutions optimal for the follower, adding constraint (6.10) to the leader will not change the bilevel feasible set.  $\square$

Since Theorem 6.7 allows us to introduce equation (6.10) into (2.4), all occurrences of  $y_i$  can now be substituted through this equation as shown in Remark 6.5.

A simple example where we can apply this technique:

**Example 6.8.**

$$\begin{aligned} \min_x \quad & y_2 - 4y_1, \\ \text{s.t.} \quad & x - y_1 \leq 3, \\ & \min_y \quad y_1 + y_2, \\ & \text{s.t.} \quad y_2 - y_1 \leq -2, \\ & \quad y_1 \geq 0, \\ & \quad -y_2 \leq -2x. \end{aligned}$$

Since  $y_2$  has positive follower coefficients except for the last constraint, where its coefficient is negative, we can substitute  $y_2 = 2x$ . This simplifies the problem to

$$\begin{aligned} \min_x \quad & -4y_1, \\ \text{s.t.} \quad & x - y_1 \leq 3, \\ & \min_y \quad y_1 + 2x, \\ & \text{s.t.} \quad y_1 \geq 2 + 2x, \\ & \quad y_1 \geq 0. \end{aligned}$$

The  $2x$  in the follower objective can be removed since that term is constant from the follower's point of view.

## 6.4 Parallel and Anti-Parallel Rows

In this chapter we will examine the occurrence of parallel rows in a linear bilevel program and how these can be used to simplify the program. The methods we use are

taken from [1], where they were established in a single-level context. We put them into the bilevel context which includes distinguishing between where the parallel rows occur (e.g. both in the follower, both in the leader, one and in the leader and one in the follower). In [1] they consider inequalities as well as equations for parallel columns. However, the methods applied to equations can be expressed through subsequent application of methods for inequalities.

In [7] an algorithm to efficiently detect parallel rows is presented.

**Example 6.9.** Consider the linear program given by

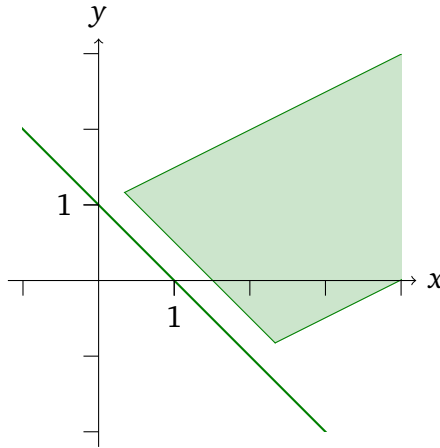
$$\min_x x \quad (6.11a)$$

$$\text{s. t. } x + y \geq 1, \quad (6.11b)$$

$$-2x - 2y \leq -3, \quad (6.11c)$$

$$y \leq 1 + \frac{x}{2}, \quad (6.11d)$$

$$y \geq -2 + \frac{x}{2}. \quad (6.11e)$$



The constraints  $x + y \geq 1$  and  $-2x - 2y \leq -3$  are parallel and by comparing the right hand sides one finds that  $x + y \geq 1$  is redundant.

The constraints  $y \leq 1 + \frac{x}{2}$  and  $y \geq -2 + \frac{x}{2}$  are anti-parallel and by comparing the right hand sides one could find that the problem is infeasible because they contradict each other, which is not the case here.

In this section we assume a linear bilevel program of the form

$$\max_{x,y} c^\top x + d^\top y \quad (6.12a)$$

$$\text{s. t. } Ax + By \leq a, \quad (6.12b)$$

$$y \in \arg \max_{\bar{y}} e^\top \bar{y} \quad (6.12c)$$

$$\text{s. t. } D\bar{y} \leq b - Cx, \quad (6.12d)$$

---

where the follower objective is maximized and the follower constraints are phrased  $D\bar{y} \leq b - Cx$  instead of  $D\bar{y} \leq b - Cx$ , since that makes it easier for us to compare leader and follower constraints.

Let  $\alpha^\top x + \beta^\top y \leq \mu$  and  $\gamma^\top x + \delta^\top y \leq \nu$  be any two constraints (or rows) of (6.12) that satisfy

$$s\alpha = \gamma, \quad s\beta = \delta$$

for some  $s \in \mathbb{R}_{\neq 0}$ . If  $s > 0$ , we call them *parallel*, otherwise *anti-parallel*.

#### 6.4.1 Presolve of Parallel Rows

If  $\alpha^\top x + \beta^\top y \leq \mu$  and  $\gamma^\top x + \delta^\top y \leq \nu$  are parallel, we get

$$\gamma^\top x + \delta^\top y \leq \nu \tag{6.13}$$

$$\iff s\alpha^\top x + s\beta^\top y \leq \nu \tag{6.14}$$

$$\iff \alpha^\top x + \beta^\top y \leq \nu/s \tag{6.15}$$

Now we can state which of the two inequalities implies the other one:

1. If  $\nu/s < \mu$  then  $\gamma^\top x + \delta^\top y \leq \nu$  is *stronger* than  $\alpha^\top x + \beta^\top y \leq \mu$ .
2. If  $\nu/s > \mu$  then  $\alpha^\top x + \beta^\top y \leq \mu$  is *stronger* than  $\gamma^\top x + \delta^\top y \leq \nu$ .
3. If  $\nu/s = \mu$  then  $\alpha^\top x + \beta^\top y \leq \mu$  and  $\gamma^\top x + \delta^\top y \leq \nu$  are equivalent.

We can now derive the following rules for handling parallel rows in a linear bilevel program:

1. Parallel leader rows:

Let  $A_i x + B_i y \leq a_i$  and  $A_j x + B_j y \leq a_j$  be two parallel leader constraints of (6.12). If either of them is stronger than the other one, then the other one can be discarded using Theorem 4.4. If they are equivalent, any one of them can be discarded using Corollary 4.4 (but obviously not both).

2. Parallel follower rows:

Let  $C_i x + D_i y \leq b_i$  and  $C_j x + D_j y \leq b_j$  be two parallel follower constraints of (6.12). If either of them is stronger than the other one, then the other one can be discarded using Corollary 4.3. If they are equivalent, any one of them can be discarded using Corollary 4.3 (but obviously not both).

3. Parallel leader and follower row:

Let  $A_i x + B_i y \leq a_i$  be a leader constraint and  $C_j x + D_j y \leq b_j$  be a follower constraint, which are parallel. If  $C_j x + D_j y \leq b_j$  is stronger than or equivalent to  $A_i x + B_i y \leq a_i$  then the latter one can be discarded using Theorem 4.4.

If, however,  $A_i x + B_i y \leq a_i$  is stronger than  $C_j x + D_j y \leq b_j$  we can argue that every point in the inducible region, i.e.  $(x, y) \in IR$ , satisfies  $A_i x + B_i y \leq a_i$  and

---

therefore  $C_j x + D_j y < b_j$ . That means that  $C_j x + D_j y < b_j$  is inactive for every point in the inducible region and  $C_j x + D_j y < b_j$  can be moved to the leader according to Theorem 4.17 without changing the inducible region. Now we can remove the new leader constraint  $C_j x + D_j y < b_j$  by applying Corollary 4.4.

### 6.4.2 Presolve of Anti-Parallel Rows

If  $\alpha^\top x + \beta^\top y \leq \mu$  and  $\gamma^\top x + \delta^\top y \leq \nu$  are anti-parallel we get

$$\gamma^\top x + \delta^\top y \leq \nu \quad (6.16)$$

$$\iff \alpha^\top x + \beta^\top y \geq \nu/s \quad (6.17)$$

and therefore

$$\nu/s \leq \alpha^\top x + \beta^\top y \leq \mu.$$

If  $\nu/s > \mu$  then the combined polyhedron, which incorporates both the leader and the follower constraints, is empty. Since the inducible region is a subset of said polyhedron, it is empty as well and (2.4) is infeasible. Therefore, anti-parallel can be used to find very basic infeasibility.

## 6.5 Rows that are Anti-Parallel to the Objective Coefficients

Alternatively to looking for two rows that are parallel, one could also look for rows that are parallel to one the objective functions. We consider the following example, which is a slightly modified version of Example 2.1:

**Example 6.10.** *The linear bilevel program*

$$\max_{x,y} 4y - x \quad (6.18a)$$

$$\text{s. t. } y \geq 1 + \frac{x}{4}, \quad (6.18b)$$

$$y \leq 5, \quad (6.18c)$$

$$y \in \operatorname{argmin}_{\bar{y}} \bar{y} \quad (6.18d)$$

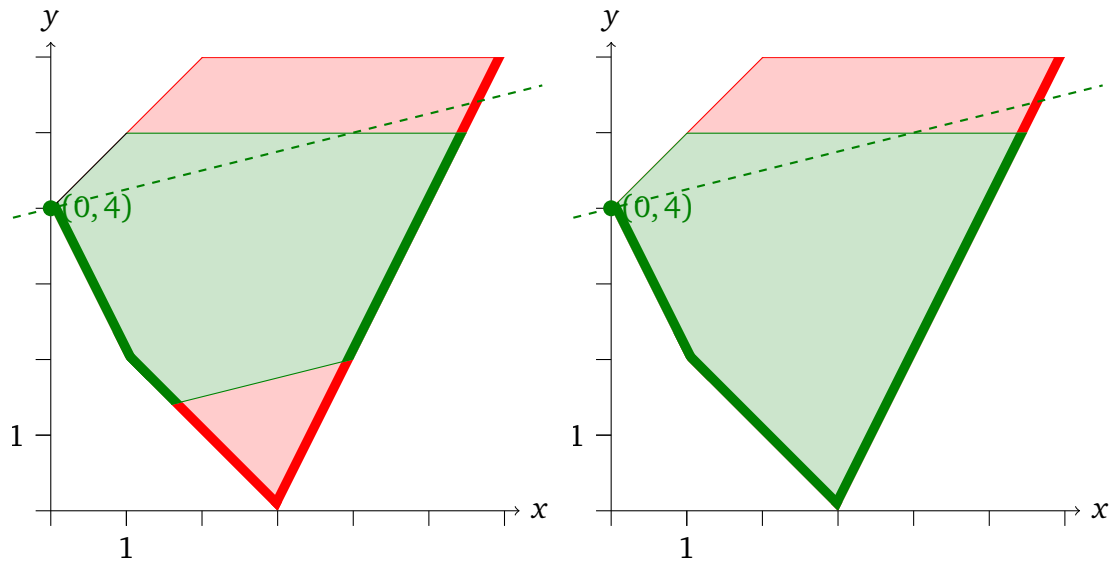
$$\text{s. t. } \bar{y} \geq 4 - 2x, \quad (6.18e)$$

$$\bar{y} \geq 3 - x, \quad (6.18f)$$

$$\bar{y} \geq -6 + 2x, \quad (6.18g)$$

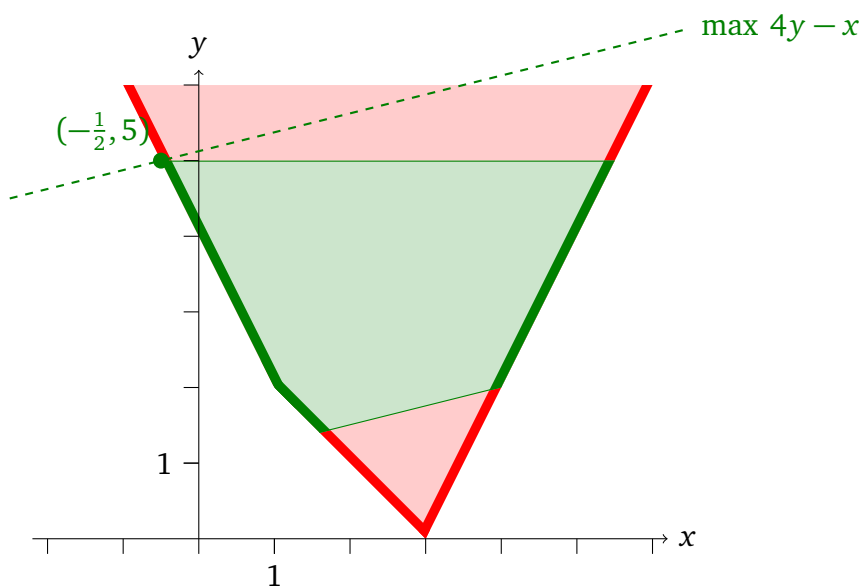
$$\bar{y} \leq 4 + x, \quad (6.18h)$$

$$\bar{y} \leq 6. \quad (6.18i)$$



The constraint  $y \geq 1 + \frac{x}{4}$  represents a lower bound for the leader objective, which will be maximized anyway. Since the whole problem is feasible,  $y \geq 1 + \frac{x}{4}$  is redundant (if the program was infeasible then removing  $y \geq 1 + \frac{x}{4}$  could make it feasible).

The follower constraints  $y \leq 4 + x$  and  $y \leq 6$  are upper bounds for the follower objective, which will be minimized anyway. Consequently, they are redundant for the follower problem as long as the follower problem is feasible. Since the feasibility of the follower problem also depends on the choice of  $x$ , these constraints might be redundant for some choices of  $x$  and make the follower problem infeasible for other choices of  $x$ . This means that we cannot remove them, but we will show that we can move them to the leader. Removing them would result in a better optimal solution:





---

In a linear bilevel program we have two objective functions a constraint can be parallel or anti-parallel to, as well as two locations (leader or follower) where that constraint can be. We will derive results for the different cases. Constraints parallel to the objective are the opposite of redundant, since they pose bounds on the objective function. We therefor focus on anti-parallel rows, which point in the opposite direction of the objective coefficients.

We will consider the linear bilevel program given by (6.12) again, where leader and follower constraints are oriented the same way and oppose the respective objectives. For starters, we will handle the simplest case, where a leader constraint is anti-parallel to the leader objective, that is

$$A_i = -\lambda c^\top, \quad B_i = -\lambda d^\top$$

for some  $i \in \{1, \dots, k\}$  and  $\lambda > 0$ . Now we assume a bilevel feasible point  $(x^*, y^*) \in IR$ . Analogously to the single-level case, we can argue that for any  $(x_0, y_0)$  with

$$A_i x_0 + B_i y_0 > a_i$$

we get

$$c^\top x_0 + d^\top y_0 = -(A_i x_0 + B_i y_0)/\lambda \tag{6.19a}$$

$$< -a_i/\lambda \tag{6.19b}$$

$$\leq -(A_i x^* + B_i y^*)/\lambda \tag{6.19c}$$

$$= c^\top x^* + d^\top y^*, \tag{6.19d}$$

which means that  $(x_0, y_0)$  has a worse objective value than  $(x^*, y^*)$ . This shows that points, which become bilevel feasible when the constraint  $A_i x + B_i y \leq a_i$  is removed from (6.12), must have a worse objective value than any existing bilevel feasible point. This allows us to remove the constraint in the process of presolve.

Note that this argument assumes that there is at least one bilevel feasible point. If there is none, removing the constraint might make an infeasible program feasible. For that reason, any solution one finds after removing the constraint has to be tested against that constraint. If it violates the constraint, the program was infeasible in the first place.

The next example shows that follower constraints anti-parallel to the leader objective cannot generally be removed without falsifying the program.

**Example 6.11.** *The linear bilevel program*

$$\max_{x,y} y \quad (6.20a)$$

$$\text{s. t. } y \leq -\frac{1}{2}x + 4, \quad (6.20b)$$

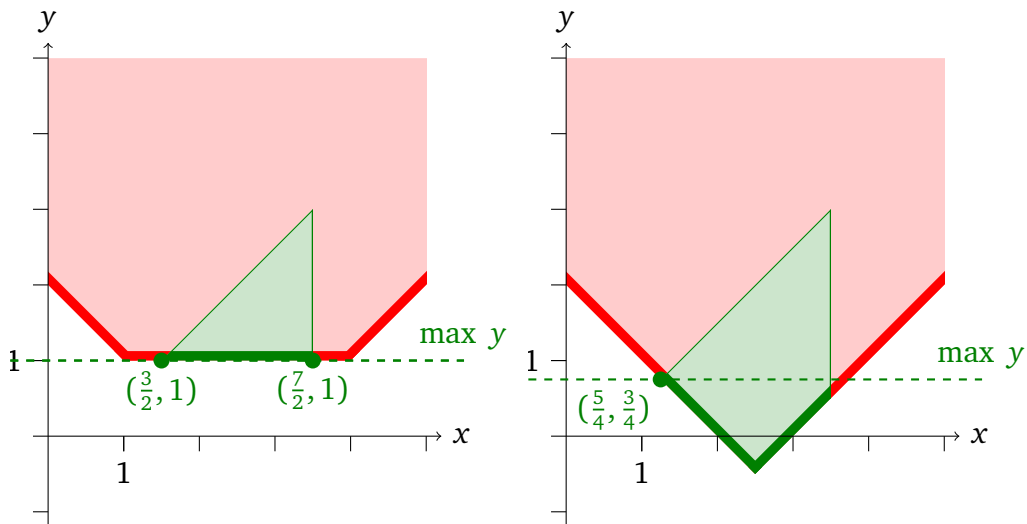
$$x \leq \frac{7}{2}, \quad (6.20c)$$

$$y \in \arg \min_{\bar{y}} \bar{y} \quad (6.20d)$$

$$\text{s. t. } \bar{y} \geq 1, \quad (6.20e)$$

$$\bar{y} \geq 2 - x, \quad (6.20f)$$

$$\bar{y} \geq -3 + x \quad (6.20g)$$



has the follower constraint  $y \geq 1$  that is anti-parallel to the leader objective. Removing it changes the optimal solution.

Now we head on to follower constraints that are anti-parallel to the follower objective, that is

$$D_j = -\lambda e^T$$

for some  $j \in \{1, \dots, l\}$  and  $\lambda > 0$ . Note that we do not require  $C_j = 0$  as technically the objective coefficients of leader  $y$  variables in the follower objective are 0. We will see that this requirement is not needed.

We consider the follower problem of (6.12) given by

$$\max_y e^T y \quad (6.21a)$$

$$\text{s. t. } Dy \leq b - Cx. \quad (6.21b)$$

---

Let  $x^* \in R^n$  so that (6.21) is feasible for  $x = x^*$ , let  $y^* \in R^m$  be an optimal solution to (6.21) for  $x = x^*$  and  $y_0 \in R^m$  with

$$D_j y_0 > b_j - C_j x^*.$$

Then we can derive (analogously to (6.19)) that

$$e^\top y_0 < e^\top y^*.$$

Again, removing constraint  $D_j y \leq b_j - C_j x$  from the follower problem only introduces worse feasible points as long as the problem was feasible for  $x = x^*$  before removing the constraint. Let  $\Psi(x)$  be the rational reaction set of (6.21) before removing the constraint and let  $\tilde{\Psi}(x)$  be the rational reaction set after removing the constraint. Then we summarize

$$\tilde{\Psi}(x) = \Psi(x) \quad \forall x \text{ where (2.5) is feasible}$$

and

$$C_j x + D_j y > b_j \quad \forall y \in \tilde{\Psi}(x) \quad \forall x \text{ where (2.5) is infeasible.}$$

This shows that all formerly bilevel feasible points are still bilevel feasible (and satisfy  $C_j x + D_j y \leq b_j$ ) and the new bilevel feasible points all violate  $D_j y \leq b_j - C_j x$ . Hence, we introduce  $C_j x + D_j y \leq b_j$  as a leader constraint to cut off these new bilevel feasible points and we ultimately did not remove the constraint but moved it from the follower to the leader.

## 6.6 Parallel Columns

The detection and handling of parallel columns is an important part of presolve, especially for mixed integer programs, since parallel columns occur more often there, as seen in [16]. First, we are going to review the core mechanic of merging parallel columns as it is described in [1]. The following example is taken from [1]:

$$\begin{aligned} \min \quad & 2x_1 + 4x_2 + x_3, \\ \text{s.t.} \quad & -x_1 - 2x_2 - x_3 \leq -10, \\ & 0 \leq x_1 \leq 3, \\ & 0 \leq x_2 \leq 4, \\ & 0 \leq x_3 \leq 5. \end{aligned}$$

Here, all coefficients of  $x_2$  are exactly twice as big as the coefficients of  $x_1$ . By substituting  $x_{\text{new}} = x_1 + 2x_2$ , the columns  $x_1$  and  $x_2$  can be merged into  $x_{\text{new}}$ . Note that the coefficients of  $x_{\text{new}}$  are the ones of  $x_1$  and the bounds of  $x_{\text{new}}$  are

$$\begin{aligned} l_{\text{new}} &= l_1 + 2l_2 = 0, \\ u_{\text{new}} &= u_1 + 2u_2 = 11. \end{aligned}$$

---

We get the new model

$$\begin{aligned}
\min \quad & 2x_{\text{new}} + x_3, \\
\text{s.t.} \quad & -x_{\text{new}} - x_3 \leq -10, \\
& 0 \leq x_{\text{new}} \leq 11, \\
& 0 \leq x_3 \leq 5.
\end{aligned}$$

Let  $M$  be an arbitrary coefficients matrix. In the case of a linear program of the form

$$\begin{aligned}
\min \quad & c^\top x \\
\text{s.t.} \quad & Ax \leq b, \\
& l \leq x \leq u,
\end{aligned}$$

this would be

$$M = \begin{pmatrix} c \\ A \end{pmatrix}.$$

Furthermore, let  $i$  and  $j$  be parallel columns in  $M$ . That means,

$$M_{\cdot,j} = \lambda M_{\cdot,i}$$

for some  $\lambda \neq 0$ . Variables  $x_i$  and  $x_j$  can now be merged through

$$\hat{x}_i = x_i + \lambda x_j$$

into a new variable with coefficients  $M_{\cdot,i}$ . The new coefficient matrix  $\widehat{M}$  is the same as  $M$  except that the  $j$ -th column is missing. Accordingly, the new variable vector  $\widehat{x}$  misses the  $j$ -th column while the  $i$ -th column represents the new merged column. Furthermore, the bounds for  $\widehat{x}_i$  are

$$\begin{aligned}
\widehat{l}_i &= l_i + \lambda l_j, \\
\widehat{u}_j &= u_i + \lambda u_j.
\end{aligned}$$

Next up we will identify equivalent solutions of the reduced and the original problem:

**Definition 6.12.** Variable vectors  $x$  and  $\widehat{x}$  for the original problem  $M$  and the reduced problem  $\widehat{M}$  are equivalent if they satisfy

$$\begin{aligned}
\widehat{x}_k &= x_k \quad \forall k \neq i, j, \\
\widehat{x}_i &= x_i + \lambda x_j, \\
l_i &\leq x_i \leq u_i, \\
l_j &\leq x_j \leq u_j.
\end{aligned}$$

For equivalent variable vectors  $x$  and  $\widehat{x}$  we get

$$\widehat{M}\widehat{x} = Mx.$$

---

This leads to the conclusion that, if  $\hat{x}$  and  $x$  satisfy Definition 6.12, evaluating the respective matrices yields the same result and each of the solutions satisfies its bounds exactly when the other solution satisfies its bounds.

Note that  $x_i$  and  $x_j$  satisfying their bounds implies that  $\hat{x}_i$  satisfies its bounds but not the other way round. That is why the bounds of  $x_i$  and  $x_j$  have to be included in the definition.

If  $M$  represents a linear program, for example, then Definition 6.12 provides an identity for solutions of the original and the merged linear program under which both solutions have the same objective value and are either both feasible or both infeasible. This ultimately proves that merging variables, as proposed in [1], delivers an equivalent linear program.

However, if  $M$  represents a linear bilevel program, its possible that one of these equivalent solutions satisfies the follower optimality constraint while the other one does not. In the following chapter we will investigate under which circumstances the aforementioned identity preserves follower optimality.

From here on,  $M$  will represent the linear bilevel program (2.4). Therefore

$$M = \begin{pmatrix} c^\top & d^\top \\ 0 & e^\top \\ A & B \\ C & D \end{pmatrix}.$$

We assume that two columns  $i$  and  $j$  of  $M$  are parallel. These columns will be merged and the resulting column will either be put into the leader or the follower, resulting in the matrix  $\widehat{M}$ .

- the leader, resulting in the matrix  $\widehat{M}$ ,
- the follower, resulting in the matrix  $\overline{M}$ .

Furthermore, let  $z = \begin{pmatrix} x \\ y \end{pmatrix}$ ,  $\widehat{z} = \begin{pmatrix} \widehat{x} \\ \widehat{y} \end{pmatrix}$  and  $\bar{z} = \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$  be variable vectors for the linear bilevel programs represented by the respective matrices. Assume that these vectors satisfy (6.12) and are therefore considered equivalent points for the respective programs. Since these variable vectors have the same objective values and share feasibility in regards to the inequalities, the question arises whether they share feasibility in regards to follower optimality. If either of  $\widehat{x}$  and  $\bar{x}$  share this feasibility with  $x$  then the respective matrix  $\widehat{M}$  or  $\overline{M}$  represents a linear bilevel program equivalent to the one represented by  $M$ .

### 6.6.1 Both Columns represent Leader Variables

We assume that two columns  $i$  and  $j$  of  $M$  are parallel. These columns will be merged and the resulting column will be put into the leader, resulting in the matrix  $\widehat{M}$ . Since both variables are leader variables, putting the new variable into the leader is the

---

obvious choice. Then, the leader decisions  $x$  and  $\hat{x}$  are equivalent if they satisfy (6.12). Since the set of follower variables stays the same, follower decisions  $y$  and  $\hat{y}$  are equivalent if  $y = \hat{y}$ . Furthermore, the follower objectives and left hand sides are identical for  $M$  and  $\hat{M}$ . While the right hand sides vary due to the merging of the leader variables, the right hand side values are still identical for equivalent leader decisions. This proves the following corollary:

**Corollary 6.13.** *When merging two parallel leader columns into a new leader column the follower problem remains identical for equivalent leader decisions.*

We can now summarize:

**Theorem 6.14.** *Two entirely parallel leader columns in a linear bilevel program can be merged into a new leader column.*

*Proof.* Since the follower problem remains the same for equivalent leader decisions, equivalent bilevel points  $z$  and  $\hat{z}$  are either both optimal solutions of the follower problem or neither of them is. They therefore share follower optimality. Beside that, they also share bilevel objective value and feasibility in the leader constraints. Therefore, the resulting linear bilevel program with the merged column is equivalent to the original one.  $\square$

As mentioned before, putting the new variable into the leader was the obvious choice and led to the desired result. Putting it into the follower instead is equal to first putting it into the leader and then moving a variable from the leader to the follower, which is a change to a bilevel problem that generally cannot be expected to sustain the same optimal solution.

### 6.6.2 Both Columns represent Follower Variables

If both variables are follower variables, putting the new variable into the follower is the obvious choice. The leader decisions  $x$  and  $\bar{x}$  are now equivalent if  $x = \bar{x}$ , while the follower decisions are equivalent if  $y$  and  $\bar{y}$  are equivalent according to Definition 6.12. Since coefficients of leader variables do not change, the right hand sides of the follower problem stay the same for equivalent choices of the leader after merging the variables.

**Theorem 6.15.** *Two entirely parallel follower columns in a linear bilevel program can be merged into a new follower column.*

*Proof.* Since the objective values, left hand side values and right hand side values, are the same in the leader and the follower for equivalent bilevel points  $z$  and  $\bar{z}$ , both of these or neither of these is feasible in the follower and they have the same objective value. Hence, either both or neither of them is an optimal solution of the follower. Furthermore, either both or neither of them is feasible to the leader and they have

---

the same leader objective value. We can conclude that bilevel optimality is sustained among equivalent bilevel points which shows that the merged linear bilevel program is equivalent to the original one.  $\square$

Again we can note that merging the new variable into a leader variable is the same as merging it into a follower variable and then moving that variable into the leader, which is generally not possible without changing the optimal solution.

### 6.6.3 Columns with just Parallel Follower Coefficients

In this section we will investigate follower columns where only the follower coefficients are parallel. Accordingly, we define

$$M = \begin{pmatrix} e^\top \\ D \end{pmatrix}.$$

We are now looking for parallel columns  $i$  and  $j$  in  $M$ . The idea is to introduce a new column  $y_{\text{new}}$  that replaces  $i$  and  $j$  in the follower, so that  $i$  and  $j$  can become leader columns. That means that  $i$  and  $j$  are being merged in the follower linear program the same way as columns are merged in a general linear program as proposed in [1]. However,  $y_i$  and  $y_j$  stay in the bilevel problem by becoming leader variables instead (they just disappear from the follower). This means that the bounds for  $y_i$  and  $y_j$  also move to the leader and a new equation  $y_{\text{new}} = \lambda y_i + y_j$  is added to the leader.

**Example 6.16.** Consider the linear bilevel program

$$\min_{x, y_1, y_2} 2y_1 + y_2 \tag{6.22a}$$

$$\text{s. t. } y_1 + y_2 \leq 5, \tag{6.22b}$$

$$0 \leq x \leq 4, \tag{6.22c}$$

$$y \in \arg \min_{\bar{y}} \bar{y}_1 + 3\bar{y}_2 \tag{6.22d}$$

$$\text{s. t. } 2\bar{y}_1 + 6\bar{y}_2 \geq x, \tag{6.22e}$$

$$\bar{y}_1 \geq 1, \quad \bar{y}_2 \geq 0. \tag{6.22f}$$

becomes

$$\min_{x, y_1, y_2, y_{new}} 2y_1 + y_2 \quad (6.23a)$$

$$\text{s. t. } y_1 + y_2 \leq 5, \quad (6.23b)$$

$$y_{new} = y_1 + 3y_2, \quad (6.23c)$$

$$0 \leq x \leq 4, \quad (6.23d)$$

$$y_1 \geq 1, \quad y_2 \geq 0, \quad (6.23e)$$

$$y_{new} \in \arg \min_{\bar{y}_{new}} \bar{y}_{new} \quad (6.23f)$$

$$\text{s. t. } 2y_{new} \geq x, \quad (6.23g)$$

$$\bar{y}_{new} \geq 1. \quad (6.23h)$$

Furthermore, either  $y_1$  or  $y_2$  can be removed. For example, all occurrences of  $y_2$  can be replaced by  $\frac{1}{3}(y_{new} - y_1)$  resulting in the following linear bilevel program:

$$\min_{x, y_1, y_{new}} 2y_1 + \frac{1}{3}(y_{new} - y_1) \quad (6.24a)$$

$$\text{s. t. } y_1 + \frac{1}{3}(y_{new} - y_1) \leq 5, \quad (6.24b)$$

$$0 \leq x \leq 4, \quad (6.24c)$$

$$y_1 \geq 1, \quad y_{new} - y_1 \geq 0, \quad (6.24d)$$

$$y_{new} \in \arg \min_{\bar{y}_{new}} \bar{y}_{new} \quad (6.24e)$$

$$\text{s. t. } 2y_{new} \geq x, \quad (6.24f)$$

$$\bar{y}_{new} \geq 1. \quad (6.24g)$$

This gets rid off the equation  $y_{new} = y_1 + 3y_2$ .

**Theorem 6.17.** A linear bilevel program where two columns of the follower problem are parallel (except for their bounds) can be reformulated so that one follower variable gets removed in return for a new leader variable.

*Proof.* To prove this we will consider the linear bilevel program in the following single-level formulation:

$$\min_{x, y} c^L \cdot x + d^L \cdot y$$

$$\text{s.t. } A^L x + B^L y \leq q^L,$$

$$y \text{ is optimal for } F(x).$$

Let  $F(x)$  be the follower problem of the linear bilevel program. This is a linear program with parameterized right hand sides. Since the merging of two columns in a linear program has no requirements on the right hand sides, the two parallel columns can be merged according to Section 6.6, resulting in a new follower variable vector  $\bar{y}$  and a new follower problem  $\bar{F}(x)$ .



---

In preparation for replacing the follower problem, we first introduce the new variable  $y_{\text{new}}$ , the constraint  $y_i + \lambda y_j = y_{\text{new}}$  and the bounds for  $y$  to the problem:

$$\begin{aligned}
& \min_{x, y, y_{\text{new}}} && c^L \cdot x + d^L \cdot y \\
& \text{s.t.} && A^L x + B^L y \leq q^L, \\
& && y_i + \lambda y_j = y_{\text{new}}, \\
& && l_i \leq y_i \leq u_i, \\
& && l_j \leq y_j \leq u_j, \\
& && y \text{ is optimal for } F(x).
\end{aligned}$$

Obviously, this does not change the optimal solution to the problem except for a redundant column. However, we now have two sub-vectors of variables  $y$  and  $\bar{y}$  that share all follower variables except for  $y_i$ ,  $y_j$  and  $y_{\text{new}}$ . However, for points that satisfy all constraints of this problem except for the optimality constraint, the vectors  $y$  and  $\bar{y}$  are equivalent according to (6.12). Therefore  $y$  is optimal for  $F(x)$  if and only if  $\bar{y}$  is optimal for  $\bar{F}(x)$ . Now we can replace the follower problem:

$$\begin{aligned}
& \min_{x, y, y_{\text{new}}} && c^L \cdot x + d^L \cdot y \\
& \text{s.t.} && A^L x + B^L y \leq q^L, \\
& && y_i + \lambda y_j = y_{\text{new}} \\
& && l_i \leq y_i \leq u_i, \\
& && l_j \leq y_j \leq u_j, \\
& && \bar{y} \text{ is optimal for } \bar{F}(x).
\end{aligned}$$

Note that if  $y_i$  and  $y_j$  both have finite lower bounds, only one of them is required, since the other one is satisfied through the lower bound for the new merged follower variable and the equation  $y_i + \lambda y_j = y_{\text{new}}$ . Same goes for the upper bounds.

Lastly, we can solve equation  $y_i + \lambda y_j = y_{\text{new}}$  for  $y_i$  or  $y_j$  and substitute the variable using Remark 6.5. □

---

## 7 Conclusion and outlook

In the course of this thesis we reviewed the existing literature on independence of irrelevant constraints of bilevel programs and we showed that for linear bilevel programs, existing theory implies that adding irrelevant constraints global optimality only for trivial problems.

We then introduced the term independence of inactive constraints and showed that this property is weaker than independence of irrelevant constraints since the latter one implies the former one but not vice versa. While independence of inactive constraints would let us remove inactive constraints, it still is a property we generally cannot expect to find in linear bilevel program. However, we proved that inactive follower constraints can always be moved to the leader. This is important, since classic solution approaches benefit from exactly this modification of the program.

The idea of moving constraints carried on to chapter 5, where we saw that information gathered by the  $K$ th-best algorithm can be exploited to find inactive constraints, and to chapter 6, where we discussed several common presolve techniques for linear programs and mixed integer programs in the light of linear bilevel programs. We generally had to distinguish between applying these techniques to leader components or to follower components. There was usually at least one case where the common presolve technique could not be applied to linear bilevel programs or just in a modified way that, for example, only allowed us to move a constraint to the leader instead of removing it.

For future studies, we see great value in a computational study of these discussed techniques, especially by evaluating the benefits of presolve for classic solution techniques for linear bilevel programs like the KKT reformulation, the  $K$ th-best algorithm or the PADM heuristic.

---

## References

- [1] ACHTERBERG, Tobias ; BIXBY, Robert E. ; GU, Zonghao ; ROTHBERG, Edward ; WENINGER, Dieter: Presolve reductions in mixed integer programming. In: *INFORMS Journal on Computing* (2019)
- [2] AMBROSIUS, Mirjam ; GRIMM, Veronika ; KLEINERT, Thomas ; LIERS, Frauke ; SCHMIDT, Martin ; ZÖTTL, Gregor: Endogenous price zones and investment incentives in electricity markets: an application of multilevel optimization with graph partitioning. 2018. – Forschungsbericht
- [3] ANDERSEN, Erling D. ; ANDERSEN, Knud D.: Presolving in linear programming. In: *Mathematical Programming* 71 (1995), Nr. 2, S. 221–245
- [4] BARD, Jonathan F.: *Practical bilevel optimization: algorithms and applications*. Bd. 30. Springer Science & Business Media, 2013
- [5] BIALAS, Wayne F. ; KARWAN, Mark H.: Two-level linear programming. In: *Management science* 30 (1984), Nr. 8, S. 1004–1020
- [6] BIXBY, Robert ; ROTHBERG, Edward: Progress in computational mixed integer programming—a look back from the other side of the tipping point. In: *Annals of Operations Research* 149 (2007), Nr. 1, S. 37
- [7] BIXBY, Robert E. ; WAGNER, Donald K.: A note on detecting simple redundancies in linear systems. In: *Operations research letters* 6 (1987), Nr. 1, S. 15–17
- [8] BREARLEY, AL ; MITRA, Gautam ; WILLIAMS, H P: Analysis of mathematical programming problems prior to applying the simplex algorithm. In: *Mathematical programming* 8 (1975), Nr. 1, S. 54–83
- [9] CAPRARA, Alberto ; CARVALHO, Margarida ; LODI, Andrea ; WOEGINGER, Gerhard J.: Bilevel knapsack with interdiction constraints. In: *INFORMS Journal on Computing* 28 (2016), Nr. 2, S. 319–333
- [10] DAXHELET, Olivier ; SMEERS, Yves: The EU regulation on cross-border trade of electricity: A two-stage equilibrium model. In: *European Journal of Operational Research* 181 (2007), Nr. 3, S. 1396–1412
- [11] DEMPE, Stephan ; KALASHNIKOV, Vyacheslav ; PÉREZ-VALDÉS, Gerardo A. ; KALASHNYKOVA, Nataliya: Bilevel programming problems. In: *Energy Systems*. Springer, Berlin (2015)
- [12] DEMPE, Stephan ; LOHSE, Sebastian: Dependence of bilevel programming on irrelevant data. (2011)
- [13] DENEGRE, Scott: Interdiction and discrete bilevel linear programming. (2011)

- 
- [14] FISCHETTI, Matteo ; LJUBIĆ, Ivana ; MONACI, Michele ; SINNL, Markus: A new general-purpose algorithm for mixed-integer bilevel linear programs. In: *Operations Research* 65 (2017), Nr. 6, S. 1615–1637
- [15] FORTUNY-AMAT, José ; MCCARL, Bruce: A representation and economic interpretation of a two-level programming problem. In: *Journal of the Operational Research Society* 32 (1981), Nr. 9, S. 783–792
- [16] GAMRATH, Gerald ; KOCH, Thorsten ; MARTIN, Alexander ; MILTENBERGER, Matthias ; WENINGER, Dieter: Progress in presolving for mixed integer programming. In: *Mathematical Programming Computation* 7 (2015), Nr. 4, S. 367–398
- [17] GRIMM, Veronika ; GRÜBEL, Julia ; SCHEWE, Lars ; SCHMIDT, Martin ; ZÖTTL, Gregor: Nonconvex equilibrium models for gas market analysis: Failure of standard techniques and alternative modeling approaches. In: *European Journal of Operational Research* 273 (2019), Nr. 3, S. 1097–1108
- [18] GRIMM, Veronika ; KLEINERT, Thomas ; LIERS, Frauke ; SCHMIDT, Martin ; ZÖTTL, Gregor: Optimal price zones of electricity markets: a mixed-integer multilevel model and global solution approaches. In: *Optimization methods and software* 34 (2019), Nr. 2, S. 406–436
- [19] GRIMM, Veronika ; MARTIN, Alexander ; SCHMIDT, Martin ; WEIBELZAHN, Martin ; ZÖTTL, Gregor: Transmission and generation investment in electricity markets: The effects of market splitting and network fee regimes. In: *European Journal of Operational Research* 254 (2016), Nr. 2, S. 493–509
- [20] GRIMM, Veronika ; SCHEWE, Lars ; SCHMIDT, Martin ; ZÖTTL, Gregor: A multi-level model of the European entry-exit gas market. In: *Mathematical Methods of Operations Research* 89 (2019), Nr. 2, S. 223–255
- [21] HANSEN, Pierre ; JAUMARD, Brigitte ; SAVARD, Gilles: New branch-and-bound rules for linear bilevel programming. In: *SIAM Journal on scientific and Statistical Computing* 13 (1992), Nr. 5, S. 1194–1217
- [22] HU, Xinmin ; RALPH, Daniel: Using EPECs to model bilevel games in restructured electricity markets with locational prices. In: *Operations research* 55 (2007), Nr. 5, S. 809–827
- [23] KLEINERT, Thomas ; LABBÉ, Martine ; PLEIN, Fränk ; SCHMIDT, Martin: There's No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization. (2019)
- [24] KLEINERT, Thomas ; SCHMIDT, Martin: Computing Feasible Points of Bilevel Problems with a Penalty Alternating Direction Method. In: *INFORMS Journal on Computing* (2019)

- 
- [25] KLEINERT, Thomas ; SCHMIDT, Martin: Global optimization of multilevel electricity market models including network design and graph partitioning. In: *Discrete Optimization* 33 (2019), S. 43–69
- [26] LABBÉ, Martine ; VIOLIN, Alessia: Bilevel programming and price setting problems. In: *4OR* 11 (2013), Nr. 1, S. 1–30
- [27] MACAL, Charles M. ; HURTER, Arthur P.: Dependence of bilevel mathematical programs on irrelevant constraints. In: *Computers & operations research* 24 (1997), Nr. 12, S. 1129–1140
- [28] PINEDA, Salvador ; MORALES, Juan M.: Solving linear bilevel problems using big-ms: Not all that glitters is gold. In: *IEEE Transactions on Power Systems* 34 (2019), Nr. 3, S. 2469–2471
- [29] STACKELBERG, Heinrich v. u. a.: Theory of the market economy. (1952)
- [30] VON STACKELBERG, Heinrich: Marktform und gleichgewicht. (1934)
- [31] ZHANG, Xiang-Sun ; LIU, De-Gang: A note on the continuity of solutions of parametric linear programs. In: *Mathematical Programming* 47 (1990), Nr. 1-3, S. 143–153

## **Erklärung**

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Erlangen, den 02.03.2020

Hierher die Unterschrift